(2)

NPS-67-84-016

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

SUPERSONIC MISSILE SEEKER LENS SYSTEM
USING GRADIENT INDEX MATERIAL
FOR FIRST ELEMENT LENS

by

David S. Davidson

September 1984

Thesis Advisor: Allen E. Fuhs

Approved for public release; distribution unlimited.

Prepared for: Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Commodore R. H. Schumaker
Superintendent

David A. Schrady
Provost

This thesis prepared in conjunction with research supported in part by Defense Advanced Research Projects Agency under DARPA Order 4035.

Released as a
Technical Report by:

J. N. DYER
Dean of Science and Engineering

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER NPS-67-84-016	2. GOVT ACCESSION NO. A151 476	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Supersonic Missile Seeker Lens System Using Gradient Index Material for First Element Lens		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; September 1984
		6. PERFORMING ORG. REPORT NUMBER NPS-67-84-016
7. AUTHOR(s) David S. Davidson		8. CONTRACT OR GRANT NUMBER(s) N62271~~8~~WE40007
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE September 1984
		13. NUMBER OF PAGES 119
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) COL Rene Larriva, USMC Defense Advanced Research Prljects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DTIC COPY INSPECTED

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

cont f p 4)

include:

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Missile Sensors, Missile Aerodynamics, Projectile Aerodynamics, Gradient Refractive Index, Lens Design, Conical Optics, Conical Lens, Guided Projectile, and Ray tracing.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The design of a supersonic missile seeker lens system using an aerodynamically efficient lens, a scanning mirror and detector was accomplished through the use of ray-tracing routines. The design of the first element lens uses a gradient in refractive index to overcome the severe handicap that the pointed shape imposes. This problem has been previously solved for positive orientations of the center of symmetry of the Gradient Refractive Index (GRIN) material with respect to the lens. In this study the negative

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract continued

orientations are resolved and the lens studies for a range of orientations and strengths of gradient. The best lens is then used in conjunction with the mirror and detector in a ray-tracing scheme. Such a system was found to be feasible and worthy of further study.

Approved for public release; distribution unlimited.

SUPERSONIC MISSILE SEEKER LENS SYSTEM USING GRADIENT INDEX MATERIAL FOR FIRST ELEMENT LENS

by

David S. Davidson
Captain, Canadian Forces
B. Eng (Engineering Physics), Royal Military College, 1976

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN PHYSICS

and

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 1984

Author: D Davidson
David S. Davidson

Approved by: Allen E. Fuhs
A. E. Fuhs, Thesis Advisor

Edmund A. Milne
E. A. Milne, Second Reader

G. E. Schacher
G. E. Schacher, Chairman
Department of Physics

M. F. Platzer
M. F. Platzer, Chairman
Department of Aeronautics

J. N. Dyer
J. N. Dyer
Dean of Science and Engineering

ABSTRACT

The design of a supersonic missile seeker lens system using an aerodynamically efficient lens, a scanning mirror and detector was accomplished through the use of ray-tracing routines. The design of the first element lens uses a gradient in refractive index to overcome the severe handicap that the pointed shape imposes. This problem has been previously solved for positive orientations of the center of symmetry of the Gradient Refractive Index (GRIN) material with respect to the lens. In this study the negative orientations are resolved and the lens studies for a range of orientations and strengths of gradient. The best lens is then used in conjunction with the mirror and detector in a ray-tracing scheme. Such a system was found to be feasible and worthy of further study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

SYMBOL	FORTRAN EQUIVALENT	DEFINITION	UNITS
a	A	index function parameter	nondimensional
arg	ARG	argument of arcsine function in GRIN ray equation	nondimensional
arg_0	-	argument of initial arcsine function in GRIN ray equation	nondimensional
b	B	index function parameter	nondimensional
c	C	ray matching constant	radians
$D_{\ell m}$	D1	ray geometrical distance from lens to mirror	nondimensional
D_{nd}	D2	ray geometrical distance from mirror to detector	nondimensional
DT	DETP	position of detector along x-axis	nondimensional
e	E	GRIN ray trace constant	nondimensional
K	CK	x-direction cosine of ray from lens to mirror	nondimensional
K'	CKK	x-direction cosine of ray from mirror to detector	nondimensional
ℓ_0	-	z-direction cosine of ray at initial point	nondimensional
L	CL	y-direction cosine of ray from lens to mirror	nondimensional
L'	CLL	y-direction cosine of ray from mirror to detector	nondimensional

M	CM	z-direction cosine of ray from lens to mirror	nondimensional
M'	CMM	z-direction cosine of ray from mirror to detector	nondimensional
M_0	-	Marchand reduced integral expression	-
MP	MIRP	position of mirror pivot point on x-axis	nondimensional
n	-	index of refraction	nondimensional
n_0	-	index of refraction at initial point of GRIN ray	nondimensional
$\vec{n}$		vector normal to mirror plane	-
n_x	-	x-component of vector normal to mirror plane	nondimensional
n_y	-	y-component of vector normal to mirror plane	nondimensional
p_0	-	x-direction cosine of GRIN ray at initial point	nondimensional
q_0	-	y-direction cosine of GRIN ray at initial point	nondimensional
r	RAD	radius to ray from center of symmetry	nondimensional
r_0	RO	radius to ray at initial point	nondimensional
r'	RP	first guess radius in iteration scheme	nondimensional
r''	-	second guess radius in iteration scheme	nondimensional
r'''	-	third guess radius in iteration scheme	nondimensional
r_{fs}	-	radius to front surface	nondimensional

$\dot{r}_{fs}$	-	derivative with respect to angle of radius to front surface	nondimensional
r_{gr}	-	radius to GRIN ray	nondimensional
$\dot{r}_{gr}$	-	derivative with respect to angle of radius to GRIN ray	nondimensional
$\vec{r}_i$	-	ray vector incident to mirror	nondimensional
r_{xi}, r_{yi}, r_{zi}	-	components of ray vector incident to mirror	nondimensional
$\vec{r}_r$	-	ray vector reflected off mirror	-
r_{xr}, r_{yr}, r_{zr}	-	components of ray reflected off mirror	nondimensional
R	R	radius from x-axis to outermost point of inner surface	nondimensional
R_z	Rzero	radius to edge of GRIN material	nondimensional
U	U	angle of incident radiation for lens construction	radians
x_d	XD	x-component of ray at detector	nondimensional
x_m	XM	x-component of ray at mirror	nondimensional
x_0	XO	x-component of ray at lens inner surface	nondimensional
y_d	YD	y-component of ray at detector	nondimensional
y_m	YM	y-component of ray at mirror	nondimensional

y_0	YO	y-component of ray at lens inner surface	nondimensional
z_d	ZD	z-component of ray at detector	nondimensional
z_m	ZM	z-component of ray at mirror	nondimensional
z_0	ZO	z-component of ray lens inner surface	nondimensional
ε	EPSLON	sign function of GRIN ray equation	-
ψ	PSI	angle between GRIN ray and radial arm from center of symmetry	radians
ψ_0	PSIO	angle between GRIN ray and radial arm at initial point	radians
θ	THETA	angle from x-axis to radius	radians
θ_0	THETO	angle from x-axis to initial point	radians
θ_L	-	angle for GRIN ray part to left of part of minimum radius	radians
θ_R	-	angle for GRIN ray part to right of point of minimum radius	radians
θ'	THETAP	angle of first guess in iteration scheme	radians
θ''	-	angle to sedonc guess in iteration scheme	radians
θ'''	-	angle of third guess in iteration scheme	radians
θ_m	THETM	angle from x-axis to plane of mirror	radians

ACKNOWLEDGEMENTS

The author would like to thank Distinguished Professor Allen E. Fuhs for his guidance and perserverance during the writing of this thesis.

The author would also like to express his gratitude to his wife Johanne for her support during this period.

I. INTRODUCTION

A. OUTLINE OF LENS DESIGN TO DATE

The design of a 'pointed' aerodynamically efficient missile seeker lens has been the topic of a number of studies. Frazier [Ref. 1], and Terrell [Ref. 2], first studied the idea of using a pointed lens as a means of obtaining some degree of aerodynamic efficiency. They also introduced the idea of using Gradient Refractive Index (GRIN) material in order to overcome the difficulties in imaging imposed by the odd shape. Amichai [Ref. 3], and Carr [Ref. 4], studied the GRIN concept in more detail, Amichai by defining the outer surface as a cone and solving for the inner surface and Carr by defining the inner surface as a cone and solving for the outer. The main reference for this author's study was drawn from Carr's work and a review of that thesis would be helpful if the reader seeks a more detailed background from that given in the next few pages.

B. GRADIENT REFRACTIVE INDEX (GRIN)

Although the theory for GRIN has been developed and known for some time it was not until fairly recently that materials have become available that allow the theory to be put into practice. A gradient in index will cause light rays to bend in an otherwise homogeneous material. Thus,

the lens designer now has a new dimension of flexibility which will allow him to solve problems that could not be formerly solved. A proper selection of the manner in which the gradient varies is also important. The designer would normally select certain degrees of symmetry. Three are most common; axial, radial or spherical gradients.

GRIN material in the shape of rods now commonly use axial and radial gradients in such devices as fiber-optic cables and photocopying machines. Spherical gradients are not available because spherical gradient material has not yet been produced. Spherical gradients do offer an advantage in that they lend themselves more easily to mathematical description since closed form solutions for the ray paths can be obtained.

A good introduction to gradients and the materials currently available for production of large scale GRIN lens systems is provided by Moore [Ref. 6], and Light [Ref. 7]. These also give the reader an idea of the advantages that may be realized along with the problems associated with the design and production of such items.

C. AIM AND INTENT OF THIS STUDY

The aim of this study is to complete the design of the Gradient Index Seeker Lens (GISL) initiated by Carr. This design uses a spherical gradient and 'pointed' lens shape. This study does not determine the aerodynamic efficiency of

the design but rather ensures that the lens has a generally pointed character that will also allow some degree of imaging for off-axis objects. The spherical gradient is chosen since it seems to fit the geometrical shape of the lens and also since the three-dimensional analysis of rays is much simpler.

This design is accomplished through the use of ray tracing techniques. Thus, no actual study of the lens material nor the manner in which such a lens could be manufactured is intended. It is the hope of the author that studies of this nature will provide the impetus for the development of suitable GRIN material with spherical gradients.

Once the lens design was completed a 'best' lens was chosen and a simple detector system applied to it to see if precise definition of off-axis targets is possible.

II. GRIN LENS DESIGN

A. GRIN LENS DESIGN METHODOLOGY

The method by which the GRIN Lens was designed in Carr's thesis [Ref. 4], is the subject of this section. A cross-section view of the top half of the lens is shown in Figure 2.1.

As can be seen, the lens is designed by defining the inner surface as a cone and then constructing the outer surface so that a backwards ray trace from a point source will result in parallel lines emerging. This is done by tracing rays from the back edge of the lens towards the front surface and finding the point of intersection of the ray and front surface line. The front surface line is found by using the previous ray solution as described below. The intersection of the ray and the line is determined using an iteration procedure. Once a solution is found the front surface line that will cause the ray to emerge parallel with all other rays is determined. This front surface line is then used to find a point solution for the next ray. About 1000 rays are traced starting at the outside edge of the inner cone and working inwards towards the center. Details of the iteration process is included in later sections.

Once the lens outside surface has been constructed the lens is then studied to see how well it will image three dimensional skew rays that enter the lens at varying degrees of incidence. To accomplish this, a grid is placed in front of the lens and then rays are traced from the intersection points of the grid through the lens and onto the image plane. The grid is then tilted at successive angles and the resulting 'spots' on the image plane recorded for each case. The manner in which these spots move on the image plane and their size are the major performance characteristics of the lens.

B. GRIN RAY PATH DESCRIPTION

Following Marchand [Ref. 5], GRIN ray paths in spherical gradients are found through the solution of the appropriate line integral:

$$= {}_0 + e \; {}^{r}_{r_0} \frac{dr}{r(n^2r^2 - e^2)^{1/2}} \qquad (2.1)$$

where

$$e = nr\sin \; = n_0 r_0 \sin {}_0 \qquad (2.2)$$

and

$$= \; 1$$

It should be noted that the value of is -1.0 if the radius is decreasing with angle and +1.0 if the radius is increasing with angle.

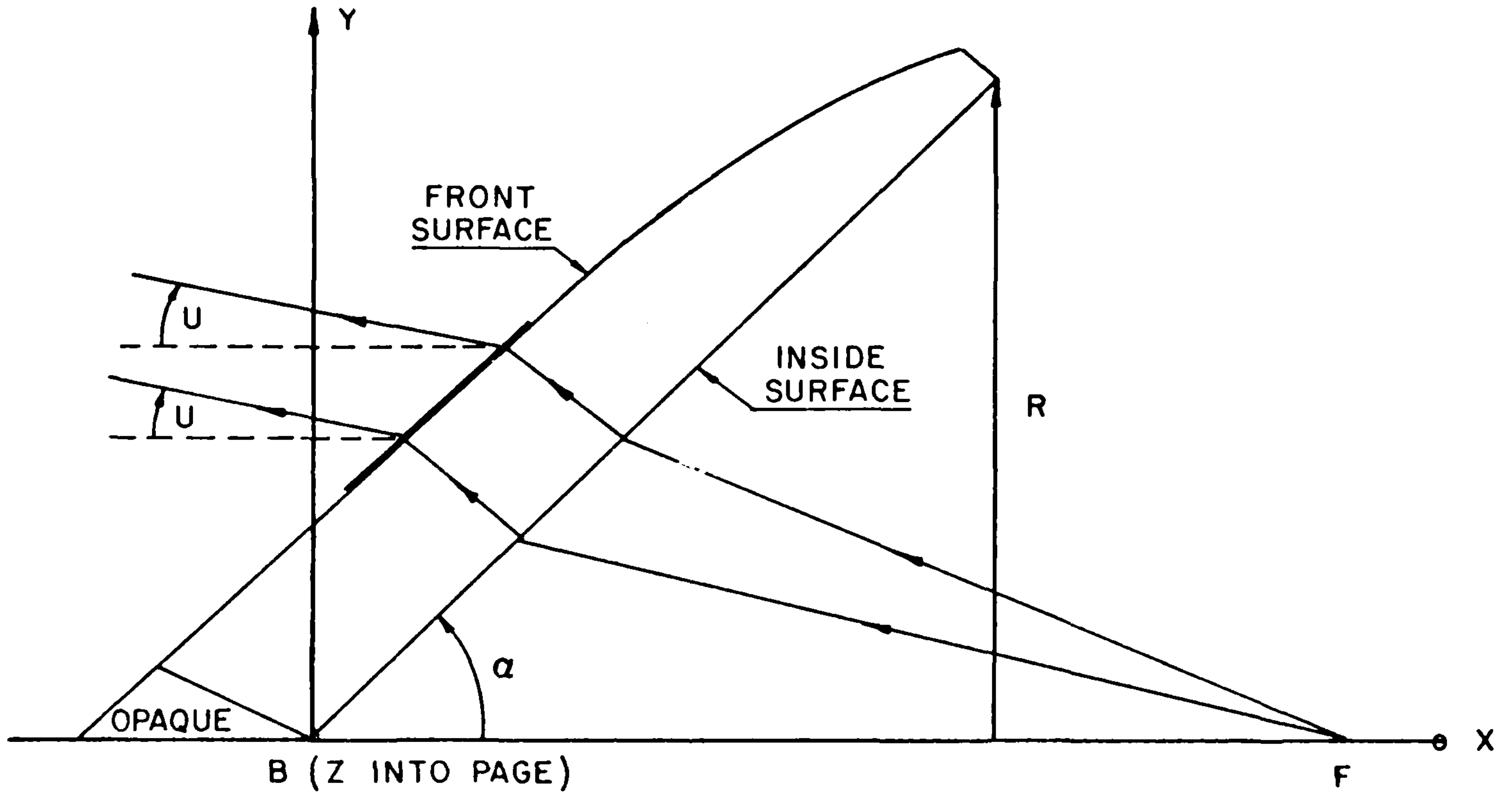

Figure 2.1 GRIN Lens Coordinate System and Parameters (from Carr).

The choice of index gradient is instrumental in allowing the integral to be solved in closed form. In this study the index is defined as:

$$n^2 = (a + br^2/R_z{}^2)$$

The constant a defines the strength of the index at the origin and the constant b defines the strength of the gradient and whether it is negative or positive.

As can be seen in Figure 2.2, the ray path is described by the constant e in conjunction with the parameter $\cdot$, the angle between the radial arm of the coordinate system and the direction of the ray at any point.

The solution of the integral in equation 2.1 results in the following expression:

$$\theta = \theta_0 - \frac{\varepsilon}{2} \{\sin^{-1}(arg_0) - \sin^{-1}(arg)\} \qquad (2.3)$$

where

$$arg_0 = \frac{a - 2e/r_0{}^2}{(a^2 + 4be^2/R_z{}^2)^{1/2}}$$

and

$$arg = \frac{a - 2e/r^2}{(a^2 + 4be^2/R_z{}^2)^{1/2}}$$

or, expressed in terms of the radius,

$$r = \frac{\sqrt{2}\,|e|}{\{a+(a^2+4be^2/R_z{}^2)\ \sin\,[-2\varepsilon(\phi-\phi_0) - \sin^{-1}(\arg_0)]\}^{1/2}} \tag{2.4}$$

C. GRIN LENS GEOMETRY

A factor which must be considered is the manner in which the lens is cut out of the GRIN material. Figure 2.2 shows an orientation of the material with respect to the lens such that the center of symmetry of the material is inside the lens. Carr has defined this as a negative value of OB, the distance from the point O to the point B. Carr was able to work out the lens problem for positive values of this parameter only, thus the major portion of this study is to develop methods that enable solutions for negative values of OB. In so doing, two factors have to be overcome:

a) ray matching - When the ray path radius initially decreases and then increases with change in angle the ray path equation must be matched at the point of minimum radius.

b) near-radial lines - Rays that show very little curvature are difficult to describe in terms of radius and angle and these cause problems in the iterative solution scheme.

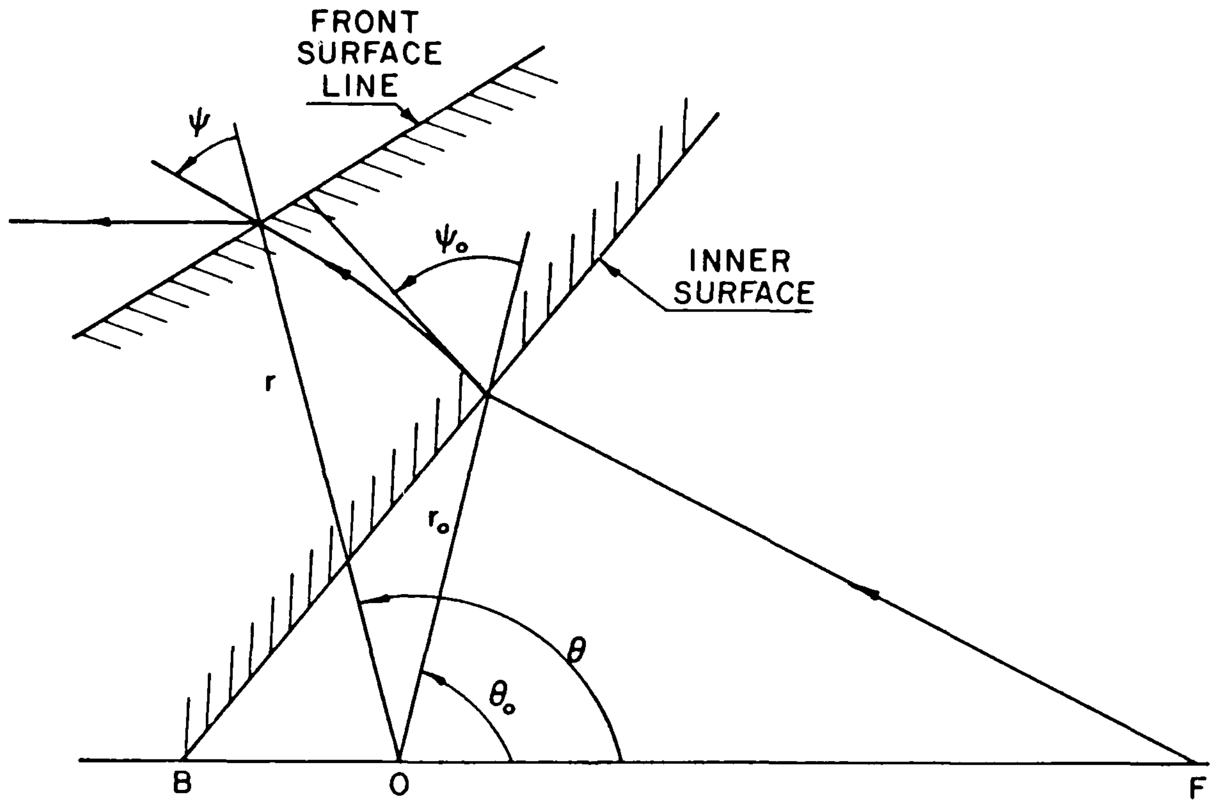

Figure 2.2 GRIN Lens and Ray Geometry.

The solution to the ray matching problem is relatively simple and is discussed in the next section. The resolution of near-radial lines is a more complex task and is covered in the following chapter.

D. RAY MATCHING

As stated previously the ray path equation changes signs as the radius of the path decreases and then increases. Such a path is shown in Figure 2.3. Simply changing sign in the equation as the point of minimum radius is passed does not produce the desired result, but rather has the effect shown in Figures 2.3 and 2.4.

The resolution of this problem is fairly simple. At the point of minimum radius the arcsine function in equation 2.3 containing this radius (arg) reduces to - /2. This is due to the fact that at this point the value of is ·/2. Thus:

$$\sin \psi = 1$$

and

$$e^2 = n^2 r^2$$

Thus:

$$\text{arg} = \frac{a-2n^2}{(a^2+4bn^2r^2/R_z^2)^{1/2}} = \frac{a-2(a+br^2/R_z^2)}{(a^2+4b(a+br^2/R_z^2)r^2/R_z^2)} = 1$$

and

$$\sin^{-1}(\arg) = -\pi/2$$

If the radius is initially decreasing as shown in Figure 2.3, we can find the angle of the minimum radius. We call this the critical angle or θ_L:

$$\theta_L = \theta_0 - \frac{\varepsilon}{2}\{\sin^{-1}(\arg_0) + \pi/2\}$$

The positive and the negative forms of the equation should both contain the point of minimum radius, in other words, by approaching the point from the left or the right the same result should be obtained. The fact that this does not happen is shown in Figure 2.3 and reveals the need to match the rays at the point of minimum radius.

This matching is achieved by simply adding a constant to either the positive or the negative form of equation 2.3 and then matching at the point of minimum radius in the following manner:

$$\theta_R = \theta_L + c$$

thus

$$-\frac{\varepsilon}{2}\{\pi/2+\sin^{-1}(\arg_0)\} = +\frac{\varepsilon}{2}\{\pi/2+\sin^{-1}(\arg_0)\} + c$$

The constant c can then be found:

$$c = -\varepsilon\{\pi/2+\sin^{-1}(\arg_0)\}$$

This constant must now be applied for all points past the point of minimum radius; in our example all points in

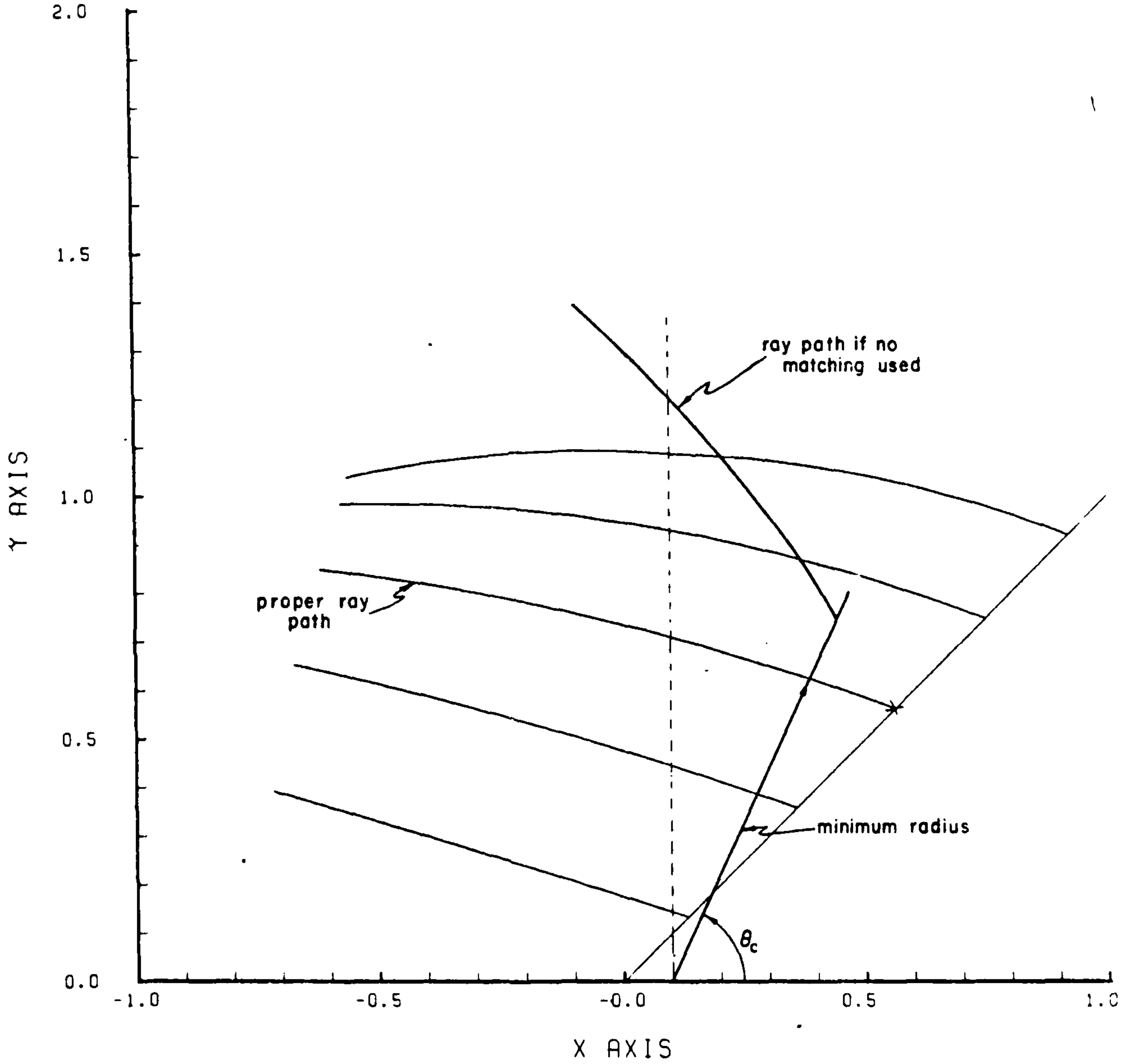

Figure 2.3 Ray Path Showing Improper Matching and Minimum Radius.

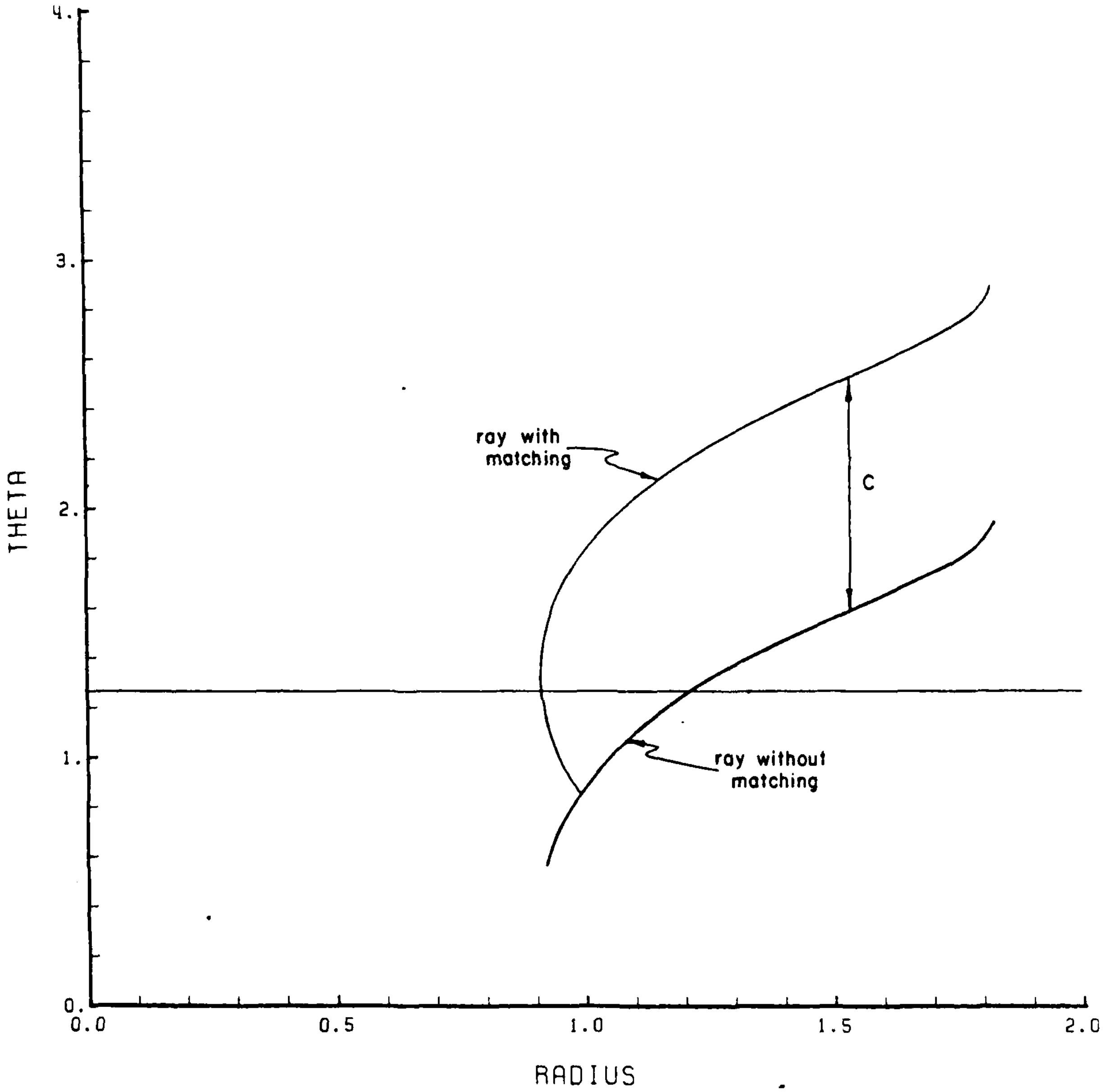

Figure 2.4 Ray Path Radius and Angle.

the increasing radius portion of the ray must be found using the form:

$$\theta = \theta_0 + \frac{\varepsilon}{2} \{\sin^{-1}(arg_0) = \sin^{-1}(arg)\} + c$$

In the iteration scheme the ray path equation must be more carefully handled. First, the critical angle and the constant c must be found. Second, the angle of the ray must be checked to see if it has passed the critical angle. If it has then the constant c is added to equation 2.3 and the point recalculated. The details of the iteration methods will be described in the following chapter.

For further clarification Figure 2.5 shows the regions in which ray matching is required for a certain position of the center of symmetry. Note also that were the value of OB positive then all rays in the lens would never reach a minimum radius point. Thus ray matching is a problem unique to the negative OB case.

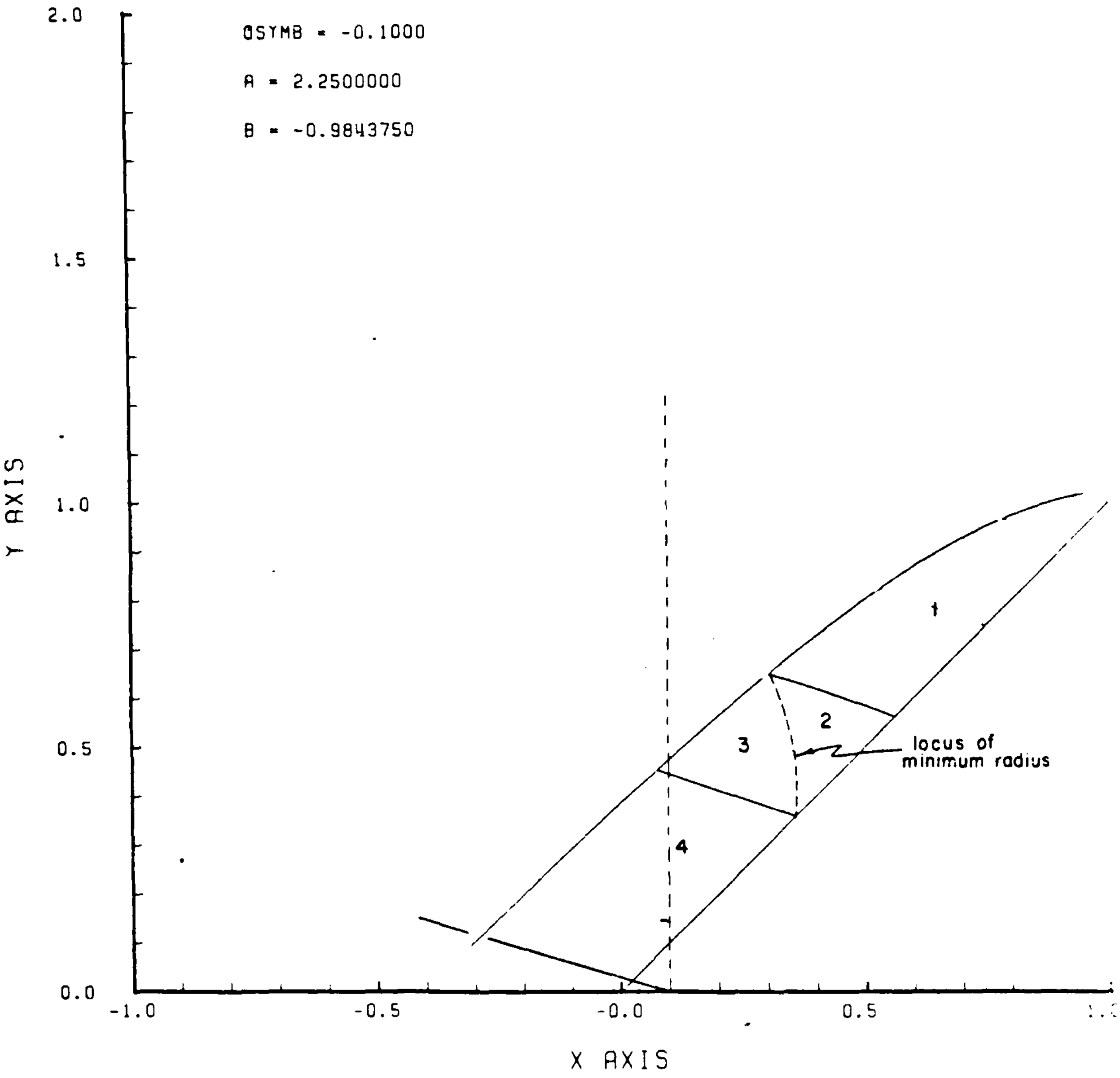

Figure 2.5 Ray Matching Regions.

III. NEAR-RADIAL LINES

A. DESCRIPTION OF DIFFICULTY

As mentioned previously, the ray paths that are nearly radial are difficult to describe mathematically in terms of radius and angle. This is due to the simple reason that a straight line originating from the center of the coordinate system has no change in angle and an undefinable change in radius.

The region in which this problem occurs in the lens design is shown in Figure 3.1. As can be seen, near-radial lines mean that the angle ψ and thus the constant e will be very small, so small as to cause equation 2.4 to approach a singularity and thus break down. In the iterative scheme, as the front surface solutions are found from the outside edge of the lens inwards, the values of ψ and e become increasingly smaller if the center of symmetry is inside the lens (OB negative). When this region is reached, equation 2.4 reduces the r_0 for all angles and futher construction of the front surface is rendered impossible. This effect is shown in Figure 3.1.

Of note is the fact that if the radial line region is passed successfully, the rays begin to curve in the opposite sense, that is they are always convex to the

radial lines in the case of a negative gradient and concave to the radial lines if a positive gradient is used.

The resolution of the near-radial lines proved to be a difficult task and became a major portion of this study. The manner in which it was overcome is the subject of the following sections.

B. PRELIMINARY WORK - IMPROVING THE ITERATION SCHEME

The first step in the resolution of the problem was to simplify the iteration scheme. In his work Carr developed a scheme based on the geometry of the rays and coordinate system [Ref. 4], which worked well for all positive values of OB (that is, for all orientations of the center of symmetry to the left of the inside surface cone. See Figure 2.2). It was decided instead to use the Newton-Raphson approach, which is simpler in concept and not dependent on geometry.

Carr used the Newton-Raphson method in the second portion of the design involving the skew ray trace, but he did not use it for the lens construction.

The manner in which this scheme works is now described so that the reader will be able to better understand the concept of how all iterative methods work for this design, as well as how the ray matching previously described fits in.

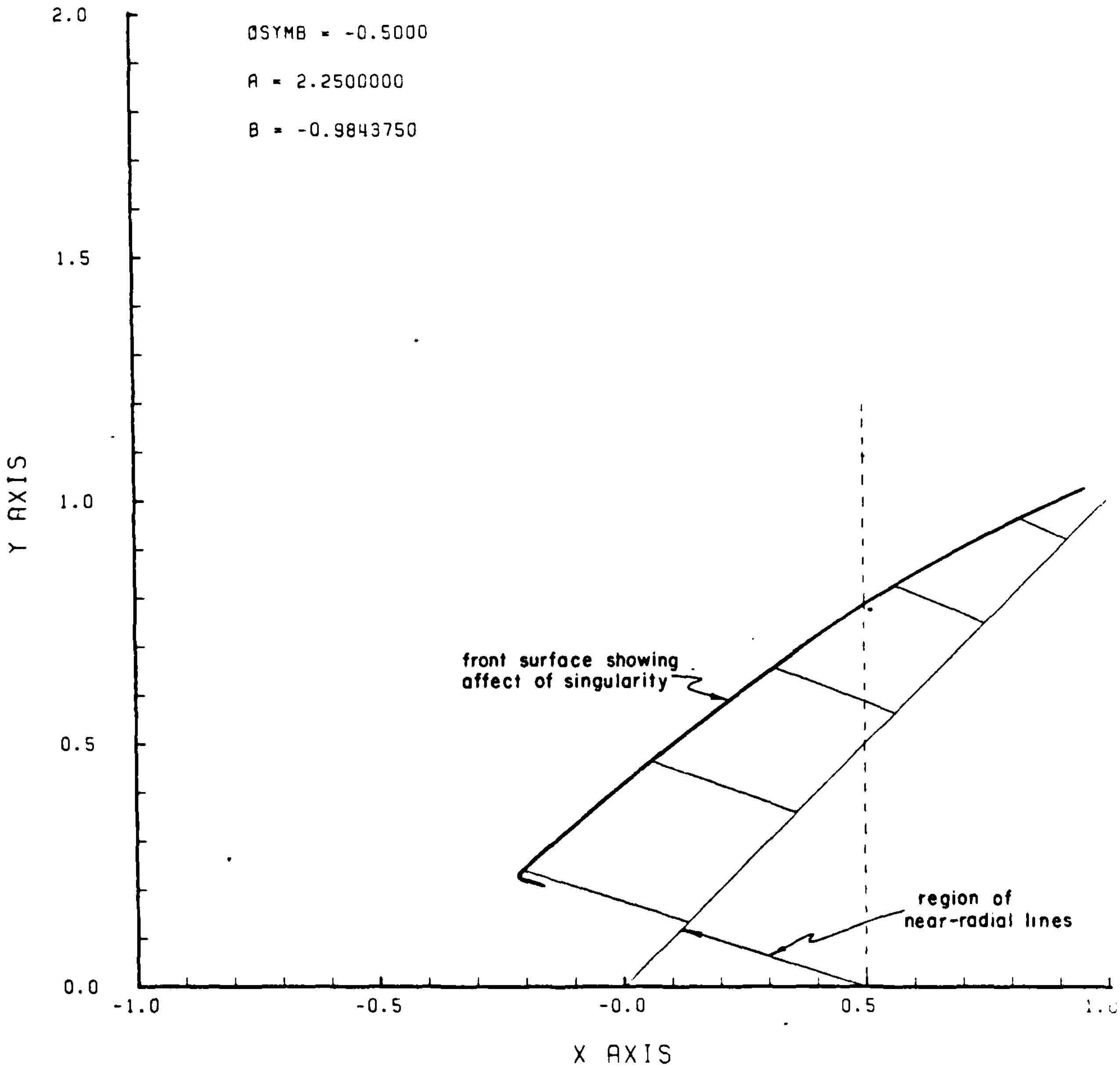

Figure 3.1 Ray Trace Showing Region of Radial Lines and Effect of Singularity.

The geometry associated with this plan is shown in Figure 3.2 and a flowchart outlining the iteration logic is shown as Figure 3.3. As can be seen with reference to the two figures, an initial point is first chosen by drawing a straight line from the inside surface intercept to the slope of the front surface obtained from a previous ray path solution. Newton-Raphson seeks to find the intercept of this front surface line with the curved ray path. This is done by using the initial angle to find the radius and derivative with respect to angle for each line and then calculating a new angle according to the relation:

$$\theta'' = \theta' - \frac{r_{fs} - r_{gr}}{\dot{r}_{fs} - \dot{r}_{gr}}$$

This new angle is then used to recalculate the radii and derivatives and thus an iterative loop if formed. Once the difference in angle is small enough the iteration is stopped and the solution considered to be found.

Ray matching is also employed in this scheme. As each new angle is calculated, it is checked to see if the critical angle has been passed. If it has, the constant c of equation 2.5 is added to the angle and thus the proper radius and derivative will be then found.

Once a solution has been found it is used to find a new front surface slope by determining that slope which will

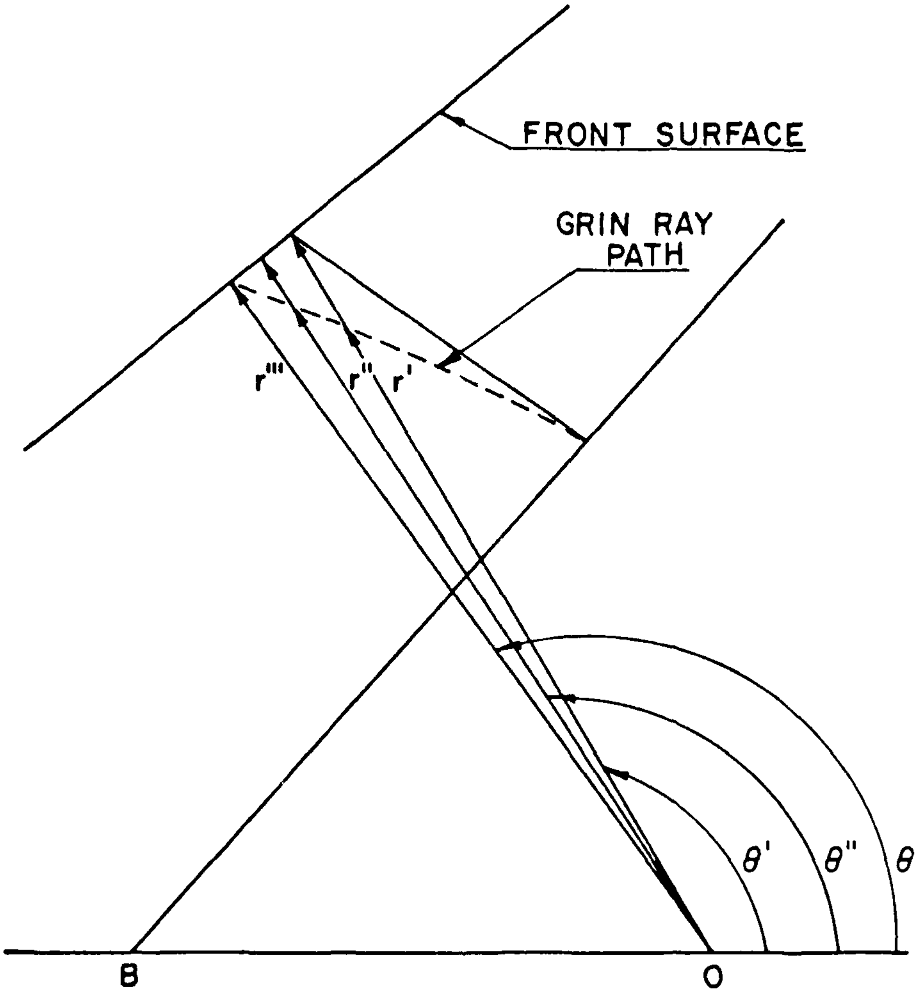

Figure 3.2 Newton-Raphson Iteration Geometry.

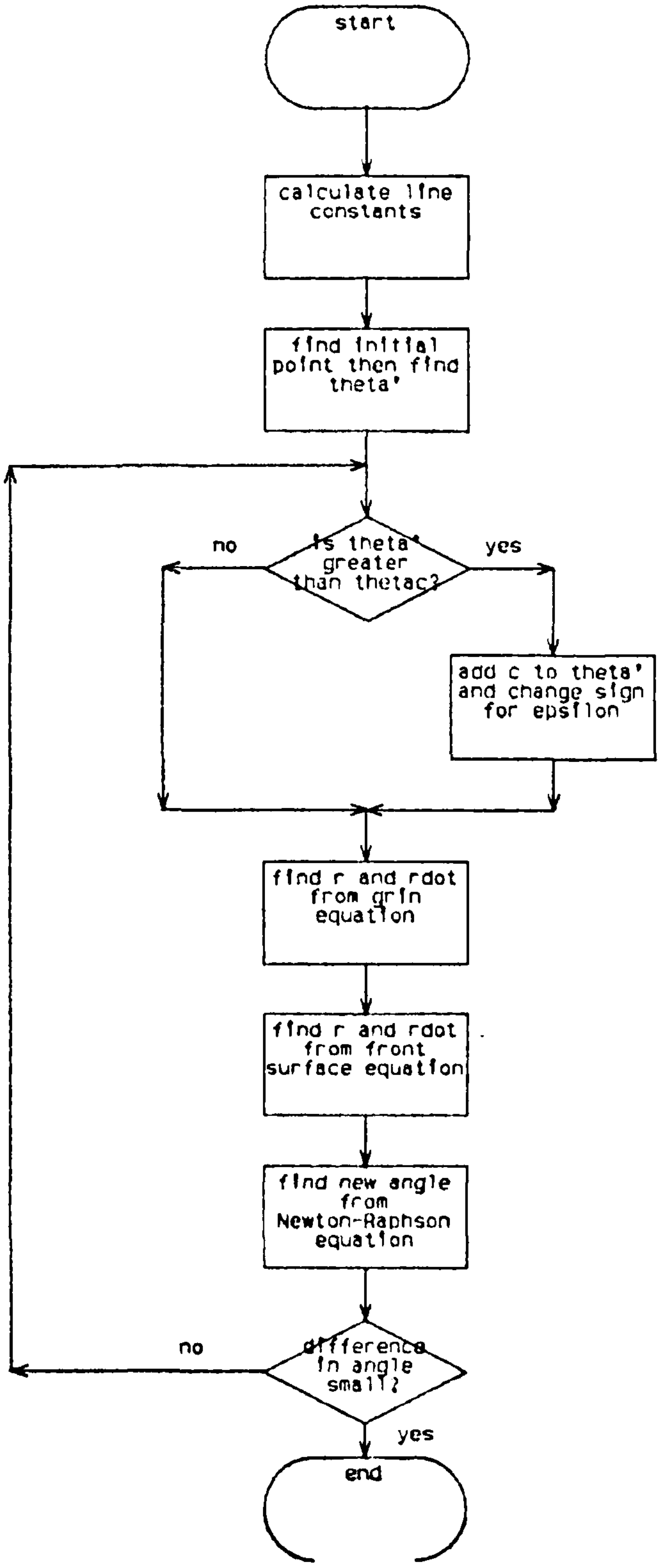

Figure 3.3 Newton-Raphson Logic Flowchart.

cause the refracted ray to emerge parallel to the axis. It is this slope that is used to find the solution for the next ray in the scheme.

When this new iterative procedure was implemented it was found to work efficiently for both the positive and negative gradients, but it did not resolve the problem of near-radial lines.

The reader should refer to Appendix A for a listing of program GISL, which is the ray tracing program originally written by Carr and modified by the author.

C. PRELIMINARY WORK - DOUBLE PRECISION

The next avenue that was explored was to place the ray tracing scheme in double precision. This was considered necessary since very small differences in angle were required in the region of the radial lines.

Once this was installed in the routine it was found to improve significantly the accuracy of the calculations and succeeded in delaying the point at which the iterations broke down. It did not solve the crux of the difficulty however.

D. PRELIMINARY WORK - CHECKING ACCURACY

At this point it was decided to develop a scheme to check the accuracy of the front surface solution. The manner in which this was done is shown in Figure 3.4

The ray trace was completed in a forward direction from the front surface to the focal point in a two-dimensional fashion. These rays were caused to originate from the mid point of each set of points that define the front surface. The slope of the front surface for each of these rays was taken to be the line drawn between the two adjacent points on the front surface. In this fashion the worst case was chosen since the skew ray trace used to determine the performance of the lens uses incident rays that can hit the front surface at any point.

These rays were then traced through the lens and back towards the focal point. The x-coordinate of the focal point was defined and the y-coordinate calculated from the ray trace to see how close it would come to the y-coordinate of the focal point originally used to start construction of the front surface. A listing of this routine is shown as Appendix B.

Another benefit of this two-dimensional ray trace was that it allowed the author to confirm the methods by which a forwards ray trace is done. In this manner ray matching and sign convention in angles was checked. Also this ray trace is different in that a solution between the ray path in the lens and the back surface, rather than the front surface, was now required in the iterative solution scheme and this could now be verified to ensure that it was properly done.

Another method of checking the accuracy of the solution would be to study the front surface smoothness in the region of near-radial lines. Although not as precise as the above method this idea allowed the author to confirm that the front surface was sufficiently smooth. By this means it was also possible to see a physical representation of the lens in the region of the singularity.

E. FIRST PROPOSED SOLUTION - STRAIGHT LINES

The first proposed method of solving the mathematical difficulty was to simply replace the near-radial lines with completely straight lines just before the singularity was reached. This could be done by following the decrease of the parameter ψ_0 until it became small enough to ensure that the lines were nearly straight.

This idea was implemented but it was found not to be sufficiently accurate. It must be emphasized that an inaccuracy in even a single point could not be tolerated since this would mean that all succeeding points in the iterative solution scheme would carry that error as well.

For these reasons the straight line solution was rejected and other means of solving the difficulty were sought.

F. SECOND PROPOSED SOLUTION - REDUCED EQUATIONS

The second method that was used to resolve the problem of near-radial lines was to reduce equation 2.4 by

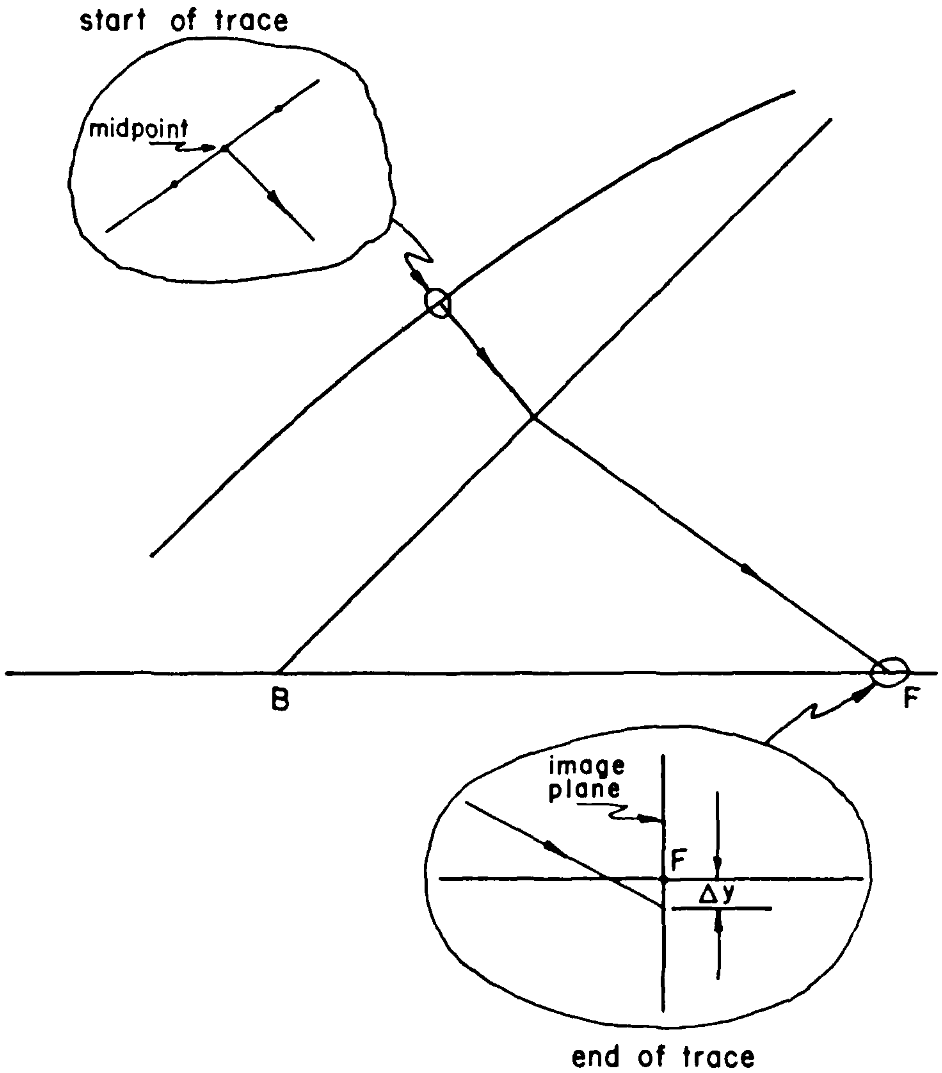

Figure 3.4 Geometry of Accuracy Check - Forwards Ray Trace.

employing small angle aproximations, binomial expansions and curve fitting. The details of how this was done follows.

Recalling equation 2.3:

$$\theta = \theta_0 - \frac{\epsilon}{2} \{\sin^{-1}(\arg_0) - \sin^{-1}(\arg)\}$$

The arguments of these arcsine functions were reworked as follows:

$$\arg_0 = \frac{a-2(a+b')\psi_0^{\,2}}{(a^2+4b'(a+b')\psi_0^{\,2})^{1/2}}$$

and

$$\arg = \frac{a-2(a+b')\psi_0^{\,2} r_0^{\,2}/r^2}{(a^2+4b'(a+b')\psi_0^{\,2})^{1/2}}$$

where

$$\psi_0 \simeq \sin\psi_0$$

$$b' = br_0^{\,2}/R_z^{\,2}$$

Note the angle ψ_0 is very small.

These arguments were then reduced using a binomial expansion:

$$\arg_0 = 1 - \frac{2(a+b')^2\psi_0^{\,2}}{a^2}$$

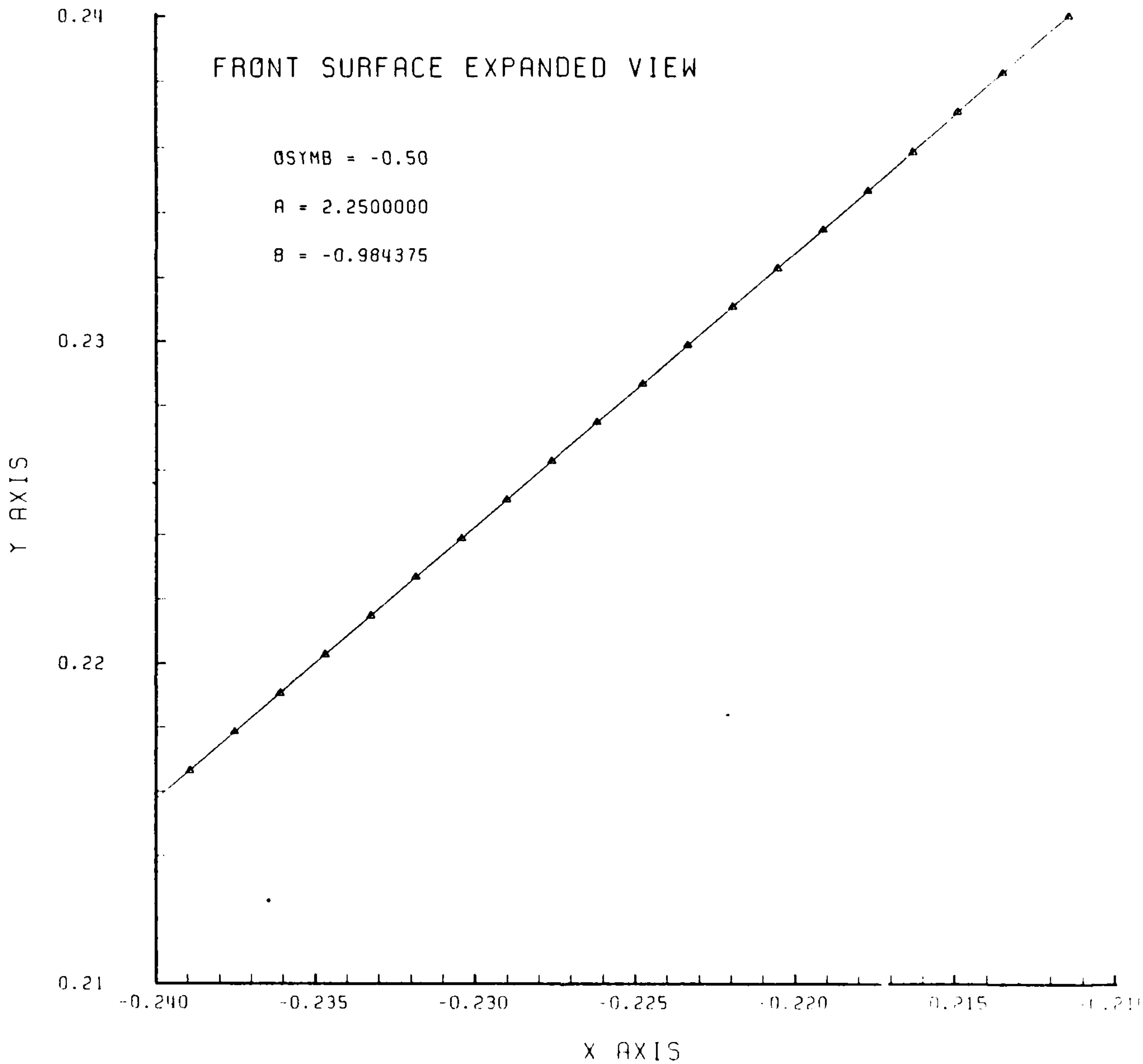

Figure 3.5 Expanded View of Front Surface -- Straight Line Method.

and

$$\arg = 1 - \frac{2(a+b')\psi_0^{\,2}\left(\frac{r_0^{\,2}}{r^2}\,a+b'\right)}{a^2}$$

Again the $\psi_0^{\,2}$ term ensures that the expansion is valid.

A function of the type arcsin(x) where x is known to be close to the value 1 can be curve fitted into the form:

$$\sin^{-1}(x) = \pi/2 - [2(1 - x)]^{1/2}$$

This equation applies to both arcsine functions as the $\psi_0^{\,2}$ term ensures that the argument is close to 1.

When placed in the equation the arguments result in the following final expression:

$$\theta = \theta_0 + \frac{\varepsilon\psi_0}{2}\left\{(a+b') - \left[(a+b')\left(\frac{r_0^{\,2}}{r^2}\,a+b'\right)\right]^{1/2}\right\}$$

In terms of radius:

$$r = r\;\frac{(a(a+b'))^{1/2}}{\left\{\left[\frac{a}{\varepsilon\psi_0}(\theta - \theta_0) - (a+b')\right]^2 - b'(a+b')\right\}^{1/2}}$$

The reduced equation was then implemented in the iterative front surface solution scheme by constantly checking it against the complete equation. When the

difference between the two became small, the reduced equation was subsequently used in solving for the front surface. The reduced equation was then turned off when the rays had progressed an equal distance on the other side of the singularity. See Figure 3.2 for clarification.

The equation worked well in resolving the front surface for the lens construction and did produce a smooth surface similar to that shown in Figure 3.5.

It did not do well when the accuracy program check was applied. One reason is that although it was easy to implement the equation for the lens construction it was not so easy to complete the forwards trace since these rays hit the front surface randomly making it difficult to know when to use the reduced equation. In practice the reduced equation could be implemented when the absolute value of the parameter ψ_0 became smaller than that used in the lens construction phase during the time that the reduced equation was used. There was then no way of telling how accurate the front surface was, in other words, there was no method to show how the rays would physically react in the region of the singularity. Thus the reduced equation method would work but it is not an entirely satisfactory solution.

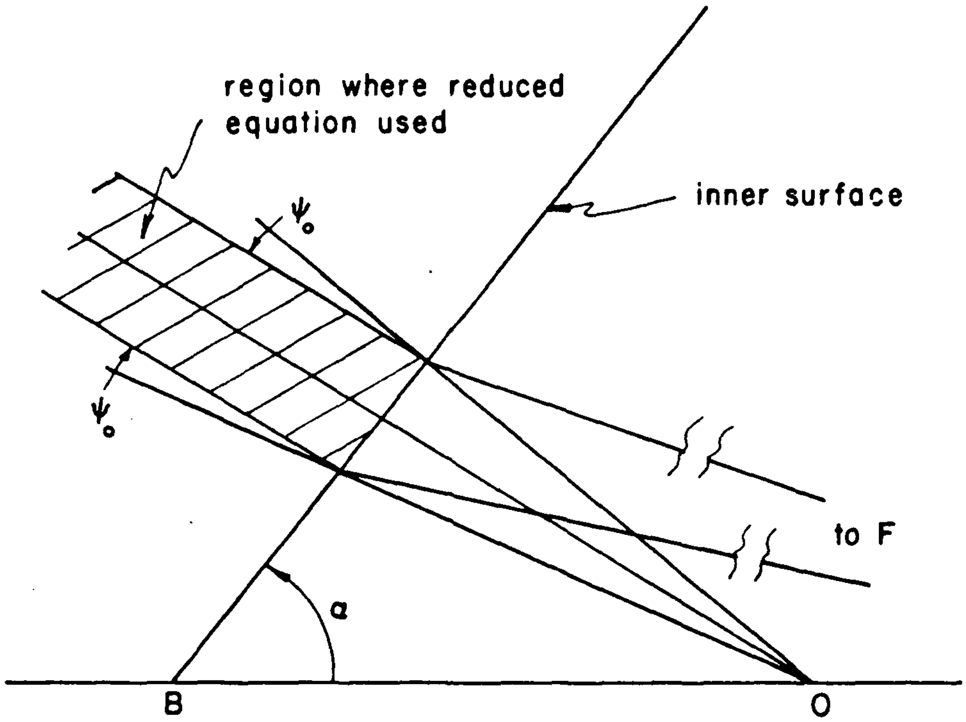

Figure 3.6 Reduced Equation Implementation.

Program CHECK was tried without using the reduced equation but disastrous results were obtained for those points in the region of the singularity.

For these reasons it was decided that another means of resolving the problem was required.

G. THIRD PROPOSED SOLUTION - MARCHAND METHOD

The third proposed solution is drawn from Marchand [Ref. 5] p. 27, in which he proposes a means of improving the ray trace formula so that nearly radial lines will be more easily described.

The method is based on redefining variables as follows:

$$\theta = eM_0 \tag{3.1}$$

where

$$M_0 = \int_{r_0}^{r} \frac{dr}{r(n^2r^2 - e^2)^{1/2}}$$

Marchand describes the ray path using the parameters α and β such that:

$$\beta = \sin\theta/\sin\theta_0$$
$$\alpha = \cos\theta - \beta\cos\theta_0 \tag{3.2}$$

This enables the coordinates of the ray to be described as:

$$x = r(\alpha\frac{x_0}{r_0} + \beta\frac{p_0}{n_0})$$

$$y = r\left(\frac{y_0}{r_0} + \frac{q_0}{n_0}\right)$$

$$z = r\left(\frac{z_0}{r_0} + \frac{0}{n_0}\right)$$

Upon substitution of equations 2.2 and 3.2, equation 3.1 becomes:

$$= n_0 r_0 M_0 \text{ sinc}$$

The parameter can now be more accurately solved for small angles of using a Maclaurin series for the sinc function.

This method was then implemented in program GISL using an iteration scheme based on Cartesian rather than polar coordinates.

This method was not found to improve the accuracy of the ray trace in the region of near radial lines and was not adopted.

One reason for this may lie in the fact that in our solution to the integral in equation 2.1 the value e no longer appears at the front of the expression. Thus there is no advantage to redefining the variable as shown in equation 3.1.

H. FOURTH PROPOSED SOLUTION - ANGLE IN TERMS OF RADIUS

The fourth method and the one that proved successful involved the solution of the front surface slope and the

GRIN ray using the form of the ray trace equation shown in equation 2.3 rather than that of equation 2.4.

The reason for this choice is shown in Figure 3.7. This figure shows that when the rays are nearly radial a small change in angle will result in a huge change in radius. Thus the form of the ray tace equation using r( ) is inherently unstable. Previously we have used this form in order to make use of the Newton-Raphson technique. It was thought that if the iteration scheme could be reworked so that the form (r) of the equation could be used then better results might be obtained.

A suitable iteration method was found and is illustrated in Figures 3.8 and 3.9.

The figures show that an initial point was chosen in the same manner as before, that is by drawing a straight line from the ray refracted at the inner surface out to the front surface slope. This point provides the radius for equation 2.4 so that a new angle can be calculated. This angle is then used to find the corresponding radius out to the front surface slope. This new radius is then used in equation 2.4 to find the next angle and thus an iterative loop is formed.

Thus a simple scheme was found. It should be noted however that this method will only converge when the initial angle is greater than 90 degrees. The reason for this lies in the geometry and is shown in Figure 3.10.

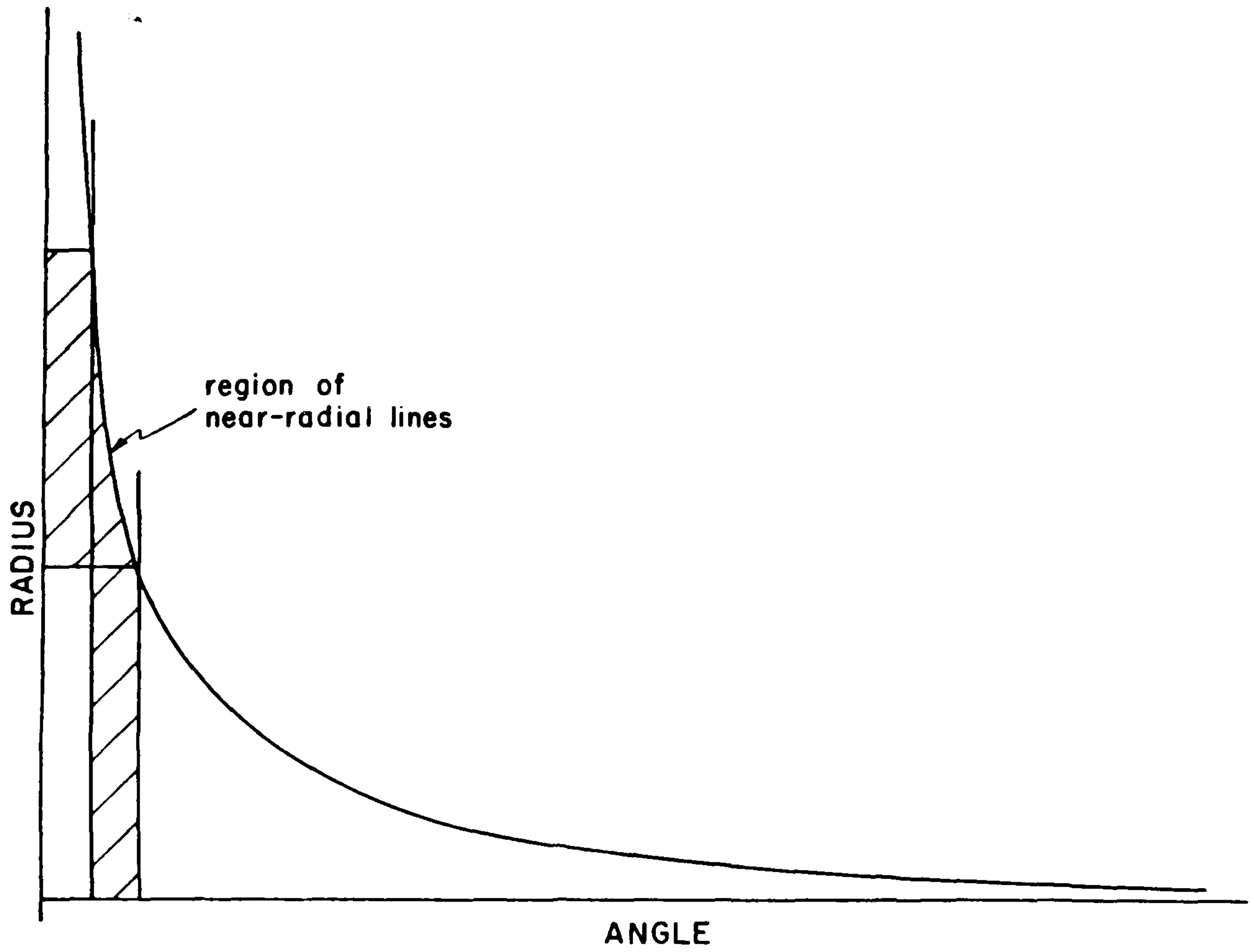

Figure 3.7 Radius and Angle Relation.

Thus the iteration will only converge when the rays are in the region of the nearly radial lines. This does not cause a difficulty since the Newton-Raphson method, modified to include ray matching, can find solutions efficiently in the region where the initial angle is less than 90 degrees.

Figure 3.10 is also drawn to show the reader an example of a forward ray trace.

This simple scheme was implemented and found to work successfully for both the lens construction and the skew ray trace; see [Ref. 8].

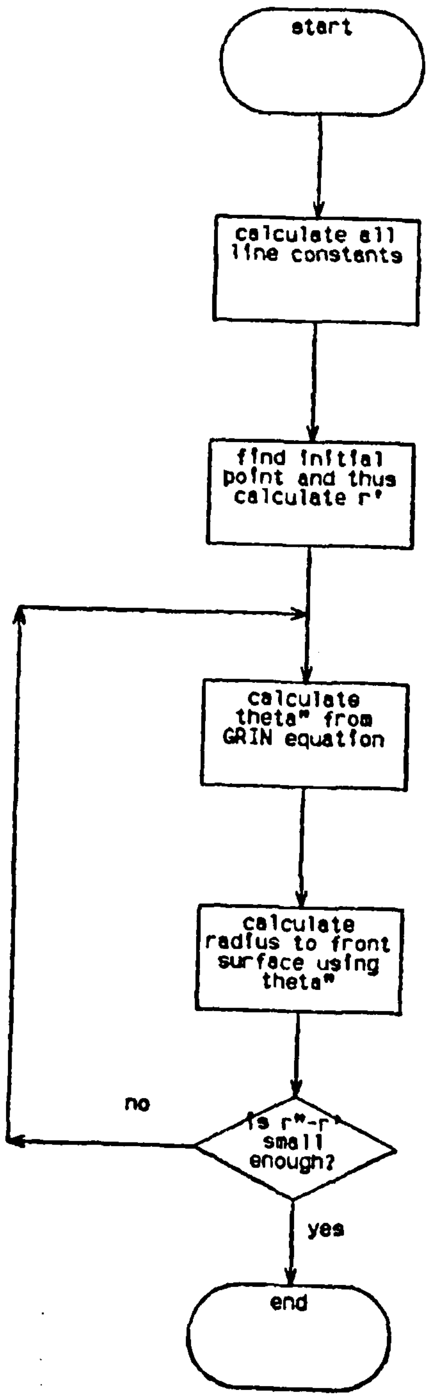

Figure 3.8 Angle in Terms of Radius Iteration Logic Flowchart.

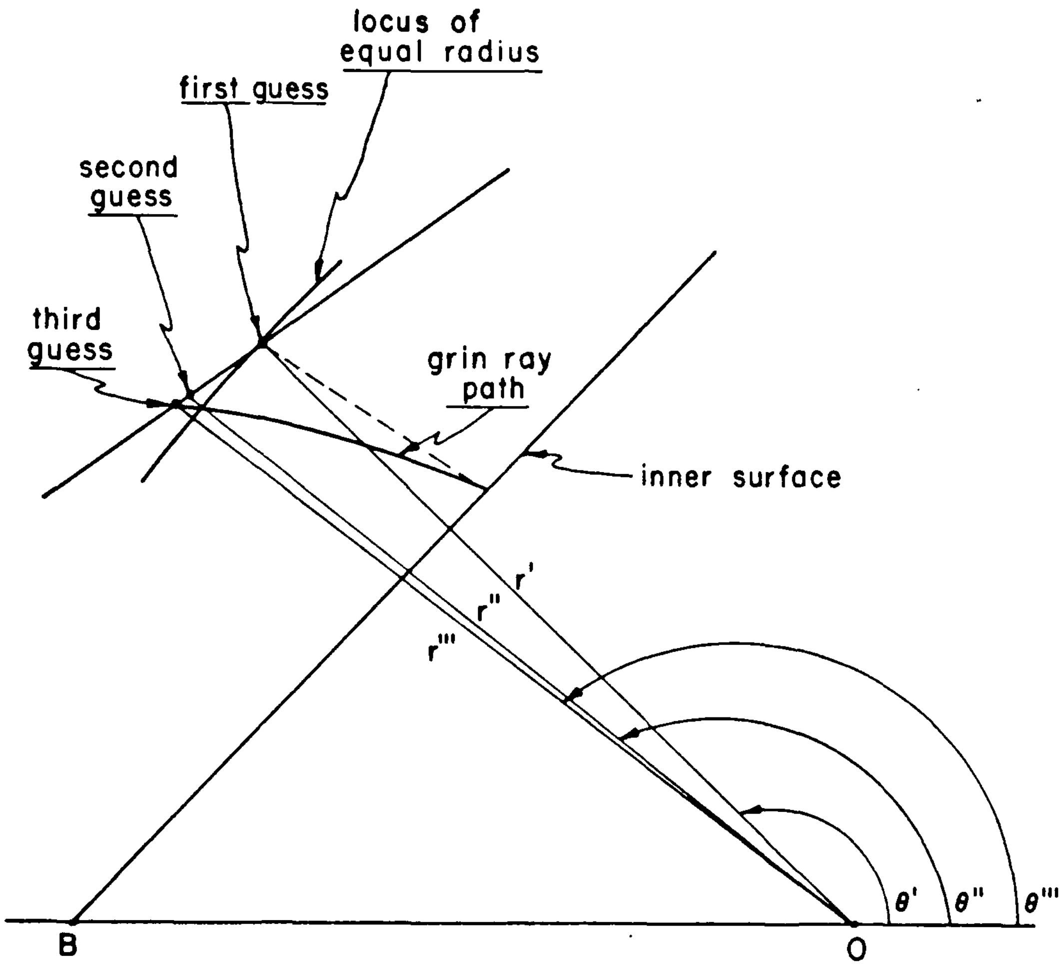

Figure 3.9 Angle in Terms of Radius Iteration Geometry.

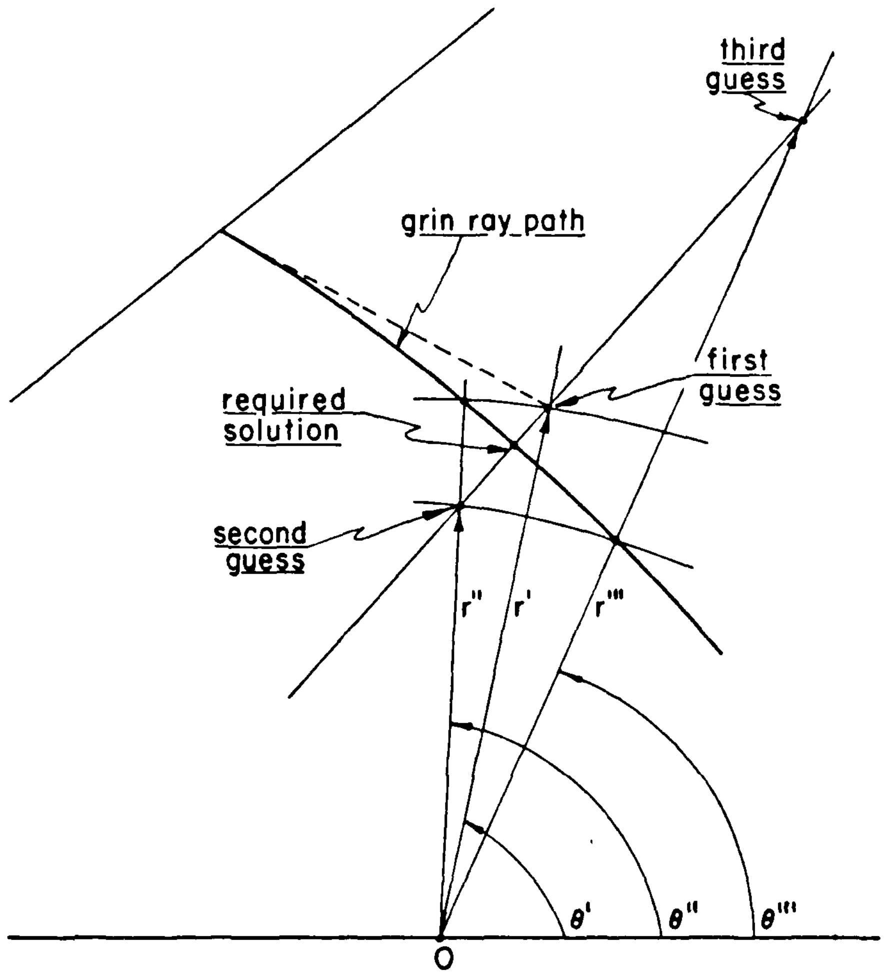

Figure 3.10 Diverging Iteration - Forward Ray Trace.

IV. GRADIENT INDEX SEEKER LENS RAY TRACE RESULTS

A. FRONT SURFACE ACCURACY

As mentioned previously, the front surface of the Gradient Index Seeker Lens was checked for accuracy by completing a ray trace from the midpoint of each set of solutions. A sample of the output from program Check is shown in Table I so that the reader will have an indication of the accuracy of the ray trace in the region of near-radial lines. Note that the most radial lines occur when the absolute value of the angle ψ_0 is smallest.

B. SPOT SIZE RESULTS

The results of program GISL showing the variation of spot size are shown in Figure 4.1. This figure shows how the spot size varies as gradient strengths and orientations of the center of symmetry are changed for a median angle of incidence of 0.3 radians.

Note that all orientations of the center of symmetry are now included. The studious reader should refer to Carr's work [Ref. 4], to compare results.

C. SELECTION OF A BEST LENS

The best lens was selected by determining those parameters that would result in a small spot size and also cause the least number of rays to be reflected or refracted

outside the missile diameter. Figure 4.1 shows that these criterion were not easily discernable and a judgemental decision was required.

The reader should note that a value of 2.25 was chosen for the parameter a. This was done since the resulting value of 1.5 for the index of refraction at the center of symmetry is common to a large number of possible lens materials. Also of note is the fact that the focal point was chosen to be 2.0 units along the x-axis from the inner surface cone. This was done since this value resulted in smaller spot sizes generally.

With these criterion in mind a value of -0.2 for OB and a 5% negative gradient were chosen as parameters for the best lens design.

A more detailed study of the lens parameters could be accomplished with use of an optimization program in which a large number of parameters could be varied in order to find the best lens. This would achieve a more satisfactory result than the one shown.

TABLE I

FRONT SURFACE ACCURACY

Line Number	Angle ψ_0 (Radians)	Error in y-coordinate	number of iterations
895	0.0411	0.0008	4
896	0.0390	0.0007	4
897	0.0369	0.0008	4
898	0.0349	0.0009	4
899	0.0328	0.0007	4
900	0.0308	0.0009	4
901	0.0288	0.0008	4
902	0.0268	0.0008	4
903	0.0247	0.0009	4
904	0.0227	0.0009	3
905	0.0207	0.0008	3
906	0.0137	0.0009	3
907	0.0167	0.0008	3
908	0.0147	0.0008	3
909	0.0127	0.0010	3
910	0.0107	0.0009	3
911	0.0087	0.0008	3
912	0.0067	0.0010	3
913	0.0048	0.0009	3
914	0.0028	0.0010	2
915	0.0008	0.0010	2
916	-0.0010	0.0010	2
917	-0.0030	0.0010	3
918	-0.0050	0.0010	3
919	-0.0069	0.0009	3
920	-0.0089	0.0011	3
921	-0.0108	0.0010	3
922	-0.0127	0.0010	3
923	-0.0147	0.0009	3
924	-0.0166	0.0011	3
925	-0.0185	0.0010	3
926	-0.0204	0.0011	4
927	-0.0223	0.0011	4
928	-0.0243	0.0010	4
929	-0.0261	0.0012	4
930	-0.0281	0.0011	4
931	-0.0300	0.0012	4
932	-0.0318	0.0012	4
933	-0.0337	0.0012	4
934	-0.0356	0.0011	4
935	-0.0375	0.0013	4

$\frac{\Delta N}{N}$ %	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
+40	.234 8,107	.225 12,105	.206 16,101	.212 17,100	.217 14,100	.219 14,100	.221 14,100	.223 12,98	.230 12,100	.233 12,101
+35	.220 12,103	.217 14,103	.205 16,101	.212 10,100	.216 15,100	.219 15,100	.221 14,100	.224 12,98	.226 13,98	.233 12,100
+30	.214 14,101	.217 14,101	.219 14,103	.211 16,101	.216 14,101	.222 14,101	.222 15,98	.224 13,98	.240 11,102	.233 12,101
+25	.220 14,101	.221 14,101	.222 14,101	.224 14,103	.215 14,101	.219 14,99	.222 14,99	.230 14,101	.238 10,103	.233 12,101
+20	.241 12,105	.230 14,103	.226 14,101	.227 14,101	.229 12,101	.234 12,103	.235 12,103	.236 12,103	.236 12,103	.249 10,105
+15	.245 12,105	.234 14,103	.234 14,103	.235 14,103	.235 12,103	.235 12,103	.235 12,103	.235 12,103	.245 10,105	.247 10,105
+10	.250 12,105	.250 12,005	.250 12,005	.249 12,105	.249 12,105	.239 12,103	.248 10,105	.247 10,105	.246 10,105	.246 10,105
+ 5	.256 12,105	.255 12,105	.255 12,105	.254 10,105	.254 12,105	.253 12,105	.251 12,105	.250 12,105	.247 12,105	.244 12,105
0	.262 12,105	.262 12,105	.261 12,105	.261 12,105	.260 12,105	.259 12,105	.257 12,105	.255 12,105	.251 12,105	.241 12,105
- 5	.240 14,107	.239 14,107	.238 14,107	.237 14,107	.237 14,107	.236 14,107	.236 14,107	.265 14,107	.256 14,105	.238 14,111
-10	.240 15,109	.239 16,109	.237 16,104	.236 16,109	.240 14,111	.239 14,111	.238 14,111	.238 14,111	.238 14,111	.247 14,111
-15	.263 17,111	.262 17,111	.261 16,111	.260 16,111	.256 16,111	.256 16,111	.256 10,111	.253 16,111	.250 14,111	.246 14,111
-20	.297 18,110	.298 18,110	.296 19,109	.292 18,109	.289 18,109	.289 18,109	.289 18,109	.285 18,109	.280 18,109	.287 18,111
-25	.294 16,111	.293 16,111	.291 16,111	.291 15,112	.290 12,114	.294 12,114	.294 12,114	.290 12,114	.280 14,112	.298 14,114

$\frac{\Delta N}{N}$ %	0	-.1	-.2	-.3	-.4	-.5	-.6	-.7	-.8	-.9	-.10
+40	.243 8,82	.216 14,97	.222 10,100	.210 18,91	.295 19,98	.202 24,92	.283 22,95	.295 24,83	.274 34,73	.309 26,77	.408 34,69
+35	.243 8,86	.217 14,97	.224 10,100	.212 18,92	.299 19,95	.250 26,95	.288 22,93	.308 28,81	.284 34,73	.324 30,71	.422 36,67
+30	.234 10,77	.218 10,99	.220 10,98	.215 20,94	.219 23,91	.218 19,96	.294 20,93	.230 30,83	.316 32,73	.342 32,68	.436 36,67
+25	.230 6,82	.220 10,99	.217 10,95	.218 18,94	.221 21,92	.220 23,93	.333 22,91	.328 30,81	.345 30,79	.379 36,72	.455 38,67
+20	.231 6,78	.217 12,99	.221 10,93	.218 18,92	.234 18,98	.253 23,89	.231 29,83	.310 33,74	.268 33,72	.432 37,72	.377 42,65
+15	.235 4,79	.220 12,98	.215 12,93	.227 16,93	.235 16,94	.249 21,94	.242 27,85	.330 33,76	.282 35,71	.466 43,65	.365 47,61
+10	.233 10,81	.239 8,100	.233 10,99	.239 10,97	.235 12,91	.253 20,84	.264 27,86	.279 37,78	.301 39,74	.350 47,66	.425 47,64
+ 5	.248 8,91	.238 10,10	.234 10,99	.233 12,97	.272 8,95	.268 26,89	.300 30,83	.309 35,73	.378 39,65	.395 47,63	.499 45,59
0	.246 12,104	.237 9,100	.290 9,100	.240 11,98	.247 12,91	.306 20,83	.340 22,79	.333 30,73	.426 39,64	.425 39,64	.588 37,60
- 5	.228 8,83	.238 9,99	.228 11,98	.230 13,94	.312 15,88	.250 21,84	.264 29,76	.385 39,71	.356 41,71	.399 47,59	.505 47,51
-10	.212 10,83	.245 8,100	.229 8,98	.226 8,92	.243 12,87	.232 25,78	.266 20,73	.401 36,53	.461 46,53	.474 41,45	.579 50,49
-15	.243 10,87	.256 6,102	.231 8,98	.270 6,97	.328 17,84	.242 10,70	.322 8,65	.287 22,55	.342 20,56	.389 36,42	.462 38,40
-20	.267 10,83	.271 4,103	.236 8,103	.237 6,86	.338 6,88	.324 8,80	.371 0,65	.390 10,52	.462 14,46	.498 16,40	.515 13,57
-25	.266 10,92	.277 7,103	.289 6,96	.274 6,92	.334 2,89	.351 2,75	.424 0,67	.418 6,58	.463 4,45	.502 10,39	.556 14,34

OSYMB

Figure 4.1 Spot Size Results - ALFAP = 0.3 Radians. Each Box Contains Spot Size, Number of Rays Outside Missile Diameter, Number of Rays on Image Plane. Total Number of Incident Rays is 135. Number of Rays Reflected Inside Lens Not Shown.

V. MIRROR AND DETECTOR SYSTEM

A. DESCRIPTION OF SYSTEM

A scanning mirror and a detector were added to the GRIN best lens in order to determine if precise definition of off-axis targets would be possible. The design of these two new elements is shown in Figure 5.1.

The mirror is made to 'nod' at a high rate and thus scans the spot across the detector. The detector consists of two sections, as shown in Figure 5.1. Thus, when the spot is scanned across the two plates a signal from each detector will be generated and will result in the form shown in Figure 5.2. This arrangement will allow an accurate measurement of the mirror angle for which the spot crosses the axis. The electronics of the detector signal processor could then be arranged to relate the mirror angle to the angle of incident radiation so that the precise line-of-sight angle of an off-axis target will be known.

Note that this is a simple detector system that will only resolve angles in one plane. A more sophisticated system along the same lines could be designed with a four-element detector and a more intricate method of scanning the spot across the detector. It was decided to keep the mirror and detector system as simple as possible since the

aim of this work is only to show that a precise angle measuring device is feasible.

B. LENS TO DETECTOR RAY TRACE

This section describes the mathematics of the ray trace from the lens to the mirror and then to the detector. A listing of program DETPLOT which accomplishes this task is included as Appendix C.

The ray trace consists of the following parts:

a. ray direction exiting lens

b. point of intersection on mirror

c. ray direction exiting mirror

d. detector ray count

1. Ray Direction Exiting Lens

The ray direction exiting the lens is derived from the direction cosines of the ray as calculated by program GISL. This data is used as an input to program DETPLOT along with the coordinates of the point at which the ray left the lens.

2. Point of Intersection on Mirror

The point of intersection of the ray and mirror may be described by the following expressions:

$$x_m = D_m K + x_0 \tag{5.1}$$

$$y_m = D_m L + y_0 \tag{5.2}$$

$$z_m = D_m M + z_0 \tag{5.3}$$

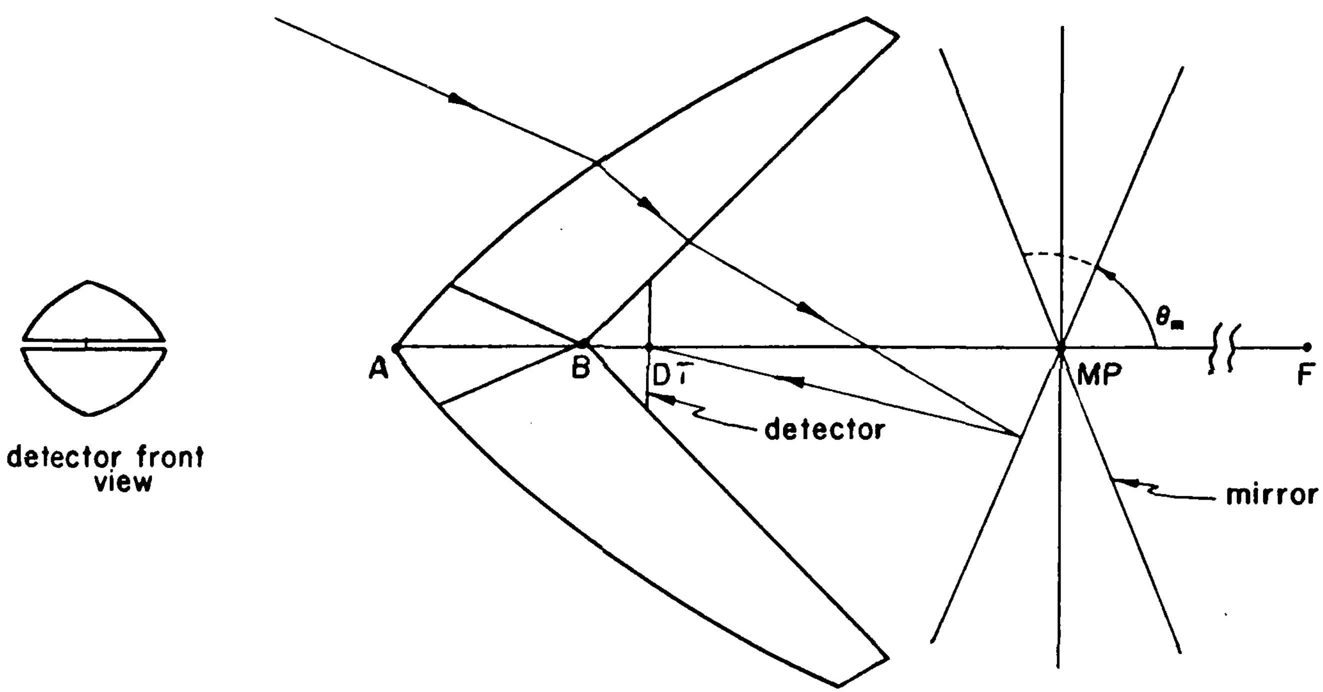

Figure 5.1 Lens, Mirror and Detector System.

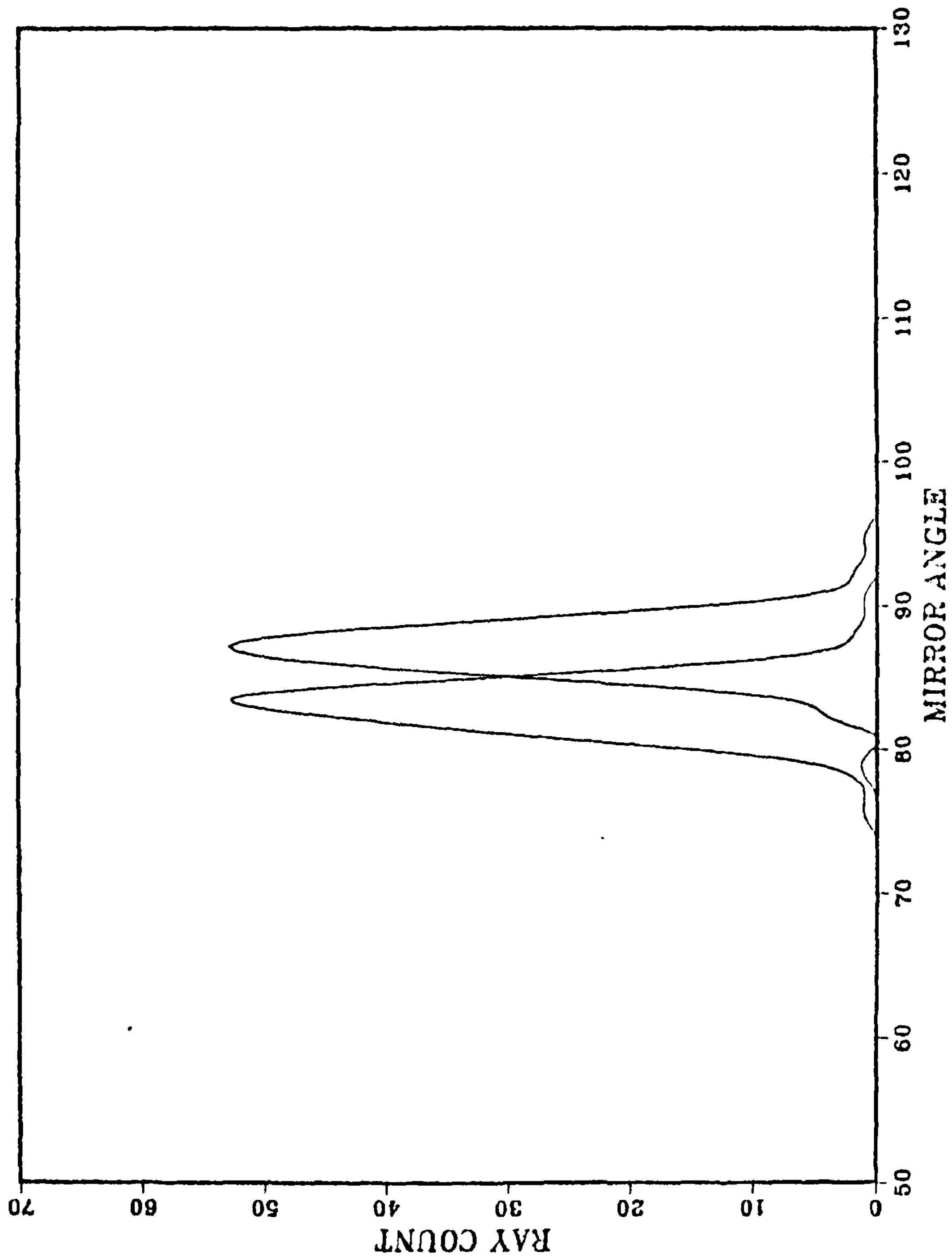

Figure 5.2 Detector Signal.

Since the surface of the mirror is tilted along the z-axis it can be described by the equations:

$$y_m = \tan\theta_M(x_m - MP) \tag{5.4}$$

$$y_m^2 \cos^2 \theta_m + z_m^2 \leq R \tag{5.5}$$

Where equation 5.4 describes a tilted plane and equation 5.5 limits the plane to the mirror diameter.

Neglecting equation 5.5 for a moment, equations 5.1, 5.2 and 5.4 may be combined to solve for $D_{\ell m}$. The result is:

$$D_{\ell m} = \frac{\tan \theta_m(x_0 - MP) - y_0}{L - K \tan \theta_m}$$

Now that $D_{\ell m}$ is known, equations 5.1, 5.2 and 5.3 may be used to find the coordinates of the mirror intersection point. These are then checked with equation 5.5 to make sure they are not outside the mirror diameter.

3. Ray Direction Exiting Mirror

The solution of the ray direction leaving the mirror is best understood vectorially. See Figure 5.3.

Reflection off a mirror is governed by two factors. Firstly, the angles of incidence and reflection must equal. Using vectors this is expressed as:

$$\vec{n} \cdot \vec{r}_i = - \vec{n} \cdot \vec{r}_r$$

In component form:

$$n_x r_{xi} + n_y r_{yi} = - n_x r_{xr} - n_y r_{yr} \tag{5.6}$$

Secondly, the ray is reflected in the same plane as the incident ray. This may be expressed vectorially by using the cross product to generate the vector $\vec{n}_p$ as shown in Figure 5.3. Thus:

$$\vec{n}_p = \vec{r}_i x \vec{n} = \vec{r}_r x \vec{n}$$

In component form:

$$\vec{e}_x(-n_y r_{zi}) + \vec{e}_y(-n_x r_{zi}) + \vec{e}_z(n_y r_{xi} - n_x r_{yi}) =$$

$$\vec{e}_x(-n_y r_{zr}) + \vec{e}_y(-n_x r_{zr}) + \vec{e}_z(n_y r_{xr} - n_x r_{yr})$$

Three equations result:

$$-n_y r_{zi} = -n_y r_{zr}$$

$$-n_x r_{zi} = -n_x r_{zr}$$

$$n_y r_{xi} - n_x r_{yi} = n_y r_{xr} - n_x r_{yr} \qquad (5.7)$$

The first two results are not unexpected as the mirror should not effect the z-component of the ray.

Equations 5.6 and 5.7 may now be combined to find the x- and y-components of the ray. When this is done the two following expressions result:

$$r_{xr} = \frac{(n_y{}^2 - n_x{}^2)}{(n_x^2 + n_y^2)} r_{xi} - \frac{2n_x n_y}{(n_x^2 + n_y^2)} r_{yi}$$

$$r_{yr} = \frac{(n_x{}^2 - n_y{}^2)}{(n_x^2 + n_y^2)} r_{yi} - \frac{2n_x n_y}{(n_x^2 + n_y^2)} r_{xi}$$

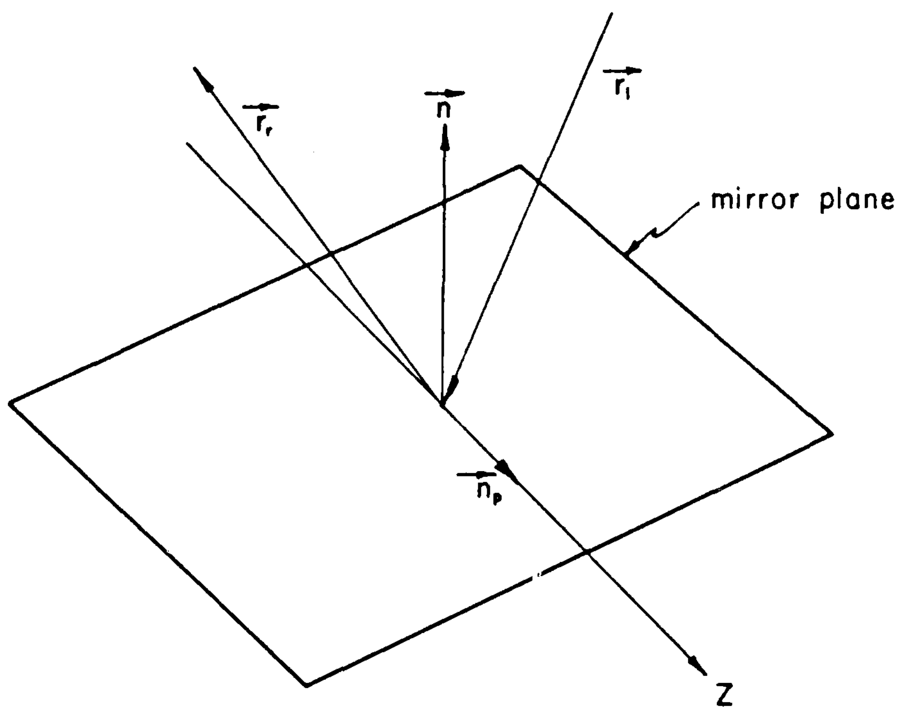

Figure 5.3 Reflection of Ray on Mirror.

Since the vector $\vec{n}$ is entirely in the x-y plane, its components may be related to mirror angle as follows:

$$n_x = -\sin\ _m$$

$$n_y = \cos\ _m$$

When the above equations, along with well known trigonometric identities are used the following expressions will be found.

$$r_{xr} = \cos(2\ _m)r_{xi} + \sin(2\ _m)r_{yi}$$

$$r_{yr} = -\cos(2\ _m)r_{yi} + \sin(2\ _m)r_{xi}$$

Thus these equations represent the change in direction for the exiting ray and form two of its direction cosines, the third being r_{zr}.

4. Ray Count on Detector

The coordinates of the ray at the detector are described by the expressions:

$$x_d = D_{nd}\ K' + x_m$$

$$y_d = D_{nd}\ L' + y_m$$

$$z_d = D_{nd}\ M' + z_m$$

Since x_d is used to define the position of the detector, it is known. Thus the value D_{nd} may be solved and the corresponding y and z-components may be found.

The ray is then checked to make sure it is within the detector diameter. If it is the y-coordinate is checked to find out if it is negative or positive, and thus

whether the ray has struck the upper or lower half of the detector. The ray count is then incremented accordingly.

Program DETPLOT traces all rays for each increment of mirror angle and thus generates plots such as that shown in Figure 5.2.

C. MIRROR AND DETECTOR SYSTEM RESULTS

Appendix D shows the results of the detector signals as a function of mirror angle at angles of incidence from 0 to 40 degrees. For these the mirror pivot point was placed at 1.4 units along the x-axis from the inner surface cone since this was the smallest distance that would still allow full movement of the scanning mirror. The focal length was adjusted to 2.8 units for the lens design calculations so that the spot would be imaged as small as possible on the detector. Figure 5.4 summarizes the results shown in Appendix D. As can be seen, the smaller angles of incidence (those in the range 0 to 16 degrees) result in a linear relation between angle of incidence and mirror tilt angle, corresponding to an on-axis spot. The larger values of the angle of incidence resulted in a more random relation between these two quantities.

Appendix D also shows that the error in determining mirror angle increases as the angle of incidence increases. This is due to the fact that each successive plot is less distinct and thus a precise measure of the angle becomes

more difficult. Although this effect shows a deterioration in the lens system performance, it is not harmful since, as the missile corrects itself, the measurement of angle will become more accurate.

For these reasons this simple system is considered to be a viable means of target detection and shows that the GISL can be a successful missile seeker lens.

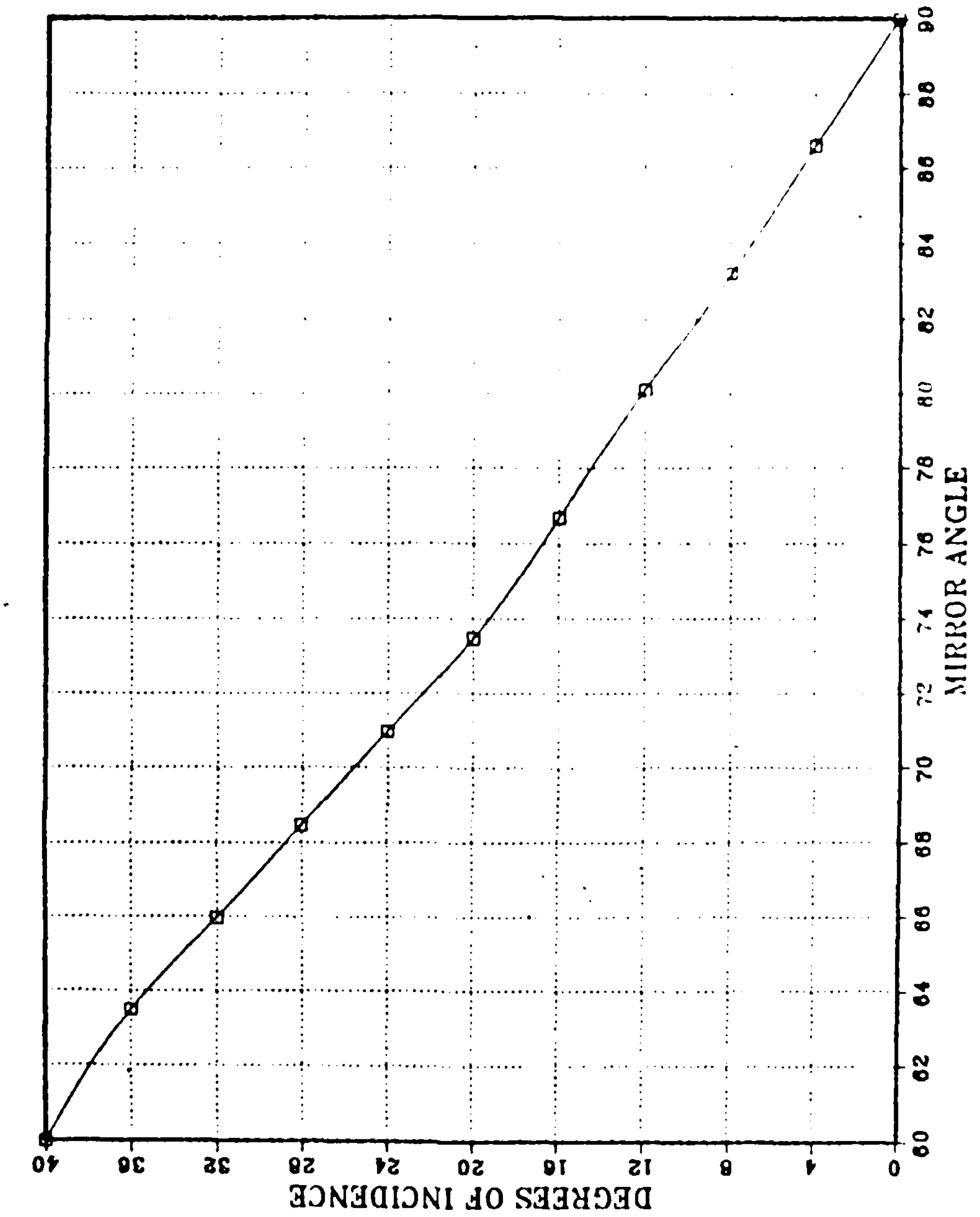

Figure 5.4 Mirror Tilt for On-Axis Spot and Angle of Incidence.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Two objectives were accomplished in the course of this study. Firstly, the Gradient Index Seeker Lens design, as initiated by Carr [Ref. 4], was completed in that all orientations of the center of symmetry with respect to the lens were made possible to describe mathematically. The results obtained for the spot size on the image plane showed trends that indicated the parameters that would produce a 'best' lens, but precise definition of these values was not possible.

Secondly, it was shown that a scanning mirror and detector system applied to the lens could be described using ray-trace techniques and also that such a system could be used to provide precise angle measurements of off-axis targets at angles less than 16 degrees. For larger values of angle, error in definition became increasingly greater.

Specifically, then, the following conclusions are made:

a) The Gradient Index Seeker Lens can be described using ray-trace techniques and can be shown to provide adequate imaging of off-axis objects.

b) A scanning mirror and detector system can be used to provide precise angle measurements.

B. RECOMMENDATIONS

The following recommendations for further study are made:

a) An optimization program should be applied to program GISL to study the variations of all important parameters.

b) A detector system capable of detection in all directions should be designed.

c) The manner in which signal processing of the detector signals can be carried out should be studied.

d) A study of Gradient Index material and the methods by which a lens could be manufactured should be undertaken.

```
C     ******************************************************************
C     *                                                                *
C     *              GRADIENT INDEX SEEKER LENS                        *
C     *              /        /     /      /                           *
C     *                                                                *
C     ******************************************************************
C     *                                                                *
C     *       THIS PROGRAM DESIGNS THE GRADIENT INDEX SEEKER LENS      *
C     *    IN TWO STAGES.   FIRST THE FRONT SURFACE IS CONSTRUCTED     *
C     *    BY TRACING RAYS FROM THE FOCAL POINT BACK THROUGH THE       *
C     *    LENS DEFINING THE FRONT SURFACE SO THAT THE RAYS EMERGE     *
C     *    PARALLEL.   SECOND A SKEW RAY TRACE IS COMPLETED IN A       *
C     *    FORWARD DIRECTION FOR ANY ANGLE OF INCIDENCE.   THE SIZE    *
C     *    OF THE RESULTING SPOT PATTERN IS RECORDED SO THAT           *
C     *    PERFORMANCE MEASUREMENTS MAY BE MADE.                       *
C     *       THIS PROGRAM WAS INITIALLY WRITTEN BY H. CARR AND        *
C     *    THEN MODIFIED BY D. DAVIDSON                                *
C     *                                                                *
C     ******************************************************************
C
C     SPECIFICATION STATEMENTS
C
      INTEGER I,J,K,L,G,H,M,N,P,FLAG(1000),Q,RAYS,S,DRAYS,ELLNUM,
     1NUMBR,CCORD,ENUM,NUMB,IANGLE,THICK,GG,HH,COUNT,SHAPE,UGT,
     1COCR(800),NUMBT,ELLIPS,SQURAY,TRIP,FILTER,CCORT,
     1FACE,FLLG,CCUNT2
      REAL DYDXN(1010),DYDXT(1010),I1P(1010),GAMMA,AB,ONP,
     1T,ALFAF,GRID,RAYY(1000),RAYZ(1000),CENI,TORBI,BOTI,TOPI,
     1RADIUS,CEN,TOP,BOT,TORB,DELTAX,DELTAY,SF,ST,RATIO,DYDXNP,LK,LL
      REAL NPI,NPJ,NPK,AO,BO,CO,DO,EC,PAR1,PAR2,NINCTY(1000),
     1ELL,ELY(7,1000),ELZ(7,1000),ELZZ,LM,ZOYC,
     1PAR3,XCIAPT(1000),YCIAPT(1000),SUM1,
     1SUM2,SUM3,SIGMAZ(10,800),SIGMAY(10,800),RMSRAD(10,800)
      REAL RCC(500),SUM4(500),FRACTN(500),SDRAD(1000),YCENTR(20,800),
     1THETAE,EPSLON,PSI,X1P,Y1P,X1HP,Y1HP,ERROR,XF,YP,ZP,RH,D2T,ABS,X1PP
      REAL Y1PP,ZETAP,BASE,BAS1,ROOO,N2X1K,N2X2K,
     1PRCNTE,PRCNTT,PRCNTC,SURFL,TSURFL,SLINCR,ARKLOS,CHEKRO,CALIBR,
     1SIGN,DIFFOR,XMAVE,XMAVET
      DOUBLE PRECISION A,E,U,F,R,ALPHA,BETA,PI,PI2,
     1X1(1010),Y1(1010),I1(1010),I2(1010),CSYMB,X2(1010),Y2(1010),X1H,
     1Y1H,I2P(1010),UP(1010),UDP(1010),N1,N2,N20,N3,E,BF,RO,RZERO,
     1DLUDP,ARG,RSINO,C,DIFF,PSIO,PSIP,ZETA,EXIT,THETAO,THETP,THETAH,
     1RAD,RADD,RDG,RDGG,RDF,RDT,EUNR,THETAP,TCRIT,THETAN,THETA,RSIN,
     1BE,SL,RF,RPF,CHECK1,CHECK2,RADH,RDCTS,RP,THTAP,BEFF,SQRE,N2X21,
     1XX,ART
      DOUBLE PRECISION XC,YO,ZO,RO,PHI,PHIP,CK,CL,CKP,CLP,CMP,CKKF,
     1CLLP,CMMP,ROX,ROY,ROZ,NPO,NPOX,NPOY,NPOZ,XX,CCK,CCL,CCM,THETA1,
     1THETA2,NUM,NUM1,NUM2,NUM3,NUM4,NUM5,NUM6,NUM7,NUM8,CSYMPH,
```

```
     1CPHX,OFFY,CPHZ,XINT,XINTD,BETAO,ALPHAO,XI,YI,ZI,A1,B1,C1,DP,
     1LKP,LLP,LMP,KKLLMM,THETAI,PHII,PHIIP,TIRA,CKPP,CLPP,CMPP
     1D1,C2,C3,XIH,YIH,ZIH,OPL(1000),YIM(1000),ZIM(1000),DIM(1000)
C
C     MAJOR VARIABLE DEFINITIONS
C
C       A          : STRENGTH OF INDEX CONSTANT
C       B          : STRENGTH OF INDEX GRADIENT
C       F          : FOCAL LENGTH
C       ALPHA      : INSIDE SURFACE CONE HALF-ANGLE
C       U          : INCIDENT RAY OFFSET ANGLE
C       I          : NUMBER OF ITERATIONS IN LENS CONSTRUCTION
C       R          : MAXIMUM INSIDE SURFACE RADIUS FROM X-AXIS
C                      NOTE THAT ALL DIMENSIONS REFERENCED TO THIS VALUE
C       T          : THICKNESS OF LENS AT EDGE
C       N1,N3      : INDICES OF REFRACTION OUTSIDE LENS
C       CSYMB      : DISTANCE FROM CENTRE OF SYMMETRY TO INSIDE SURFACE
C                      ALONG X-AXIS. POSITIVE IF O TO LEFT OF B.
C       ALFAP      : ANGLE OF INCIDENT RADIATION FOR SKEW RAY TRACE
C       GRID       : GRID INCREMENT, DEFINES NUMBER OF SKEW RAYS
C       ENUM       : NUMBER OF ELLIPSES. MUST BE INTEGER AND ODD
C       ELL        : Z-COORDINATE INCREMENT FOR ELLIPSE PLOTS
C
C
C       OUTPUT OPTIONS:
C
C       TO HAVE LENS SHAPE DATA PRINTED SET "SHAPE" TO 1. OTHERWISE
C       SET "SHAPE"=0.
        SHAPE=0
C
C       TO HAVE SKEW RAY AND MIRROR IMAGE SKEW RAY DATA PRINTED SET
C       "SQURAY" TO 1. ELSE SET "SQURAY" TO 0. (INTEGER)
        SQURAY=0
C
C       TO HAVE ELLIPSE SHAPE DATA PRINTED SET "ELLIPS" TO 1. OTHERWISE
C       SET "ELLIPS"=0.
        ELLIPS=0
C
C     SET CONSTANTS
C
        F=4.0
        ALPHA=0.7853982
        U=0.000000
        I=1008
        R=1.0
        T=0.05
        N1=1.0
        N3=1.0
```

```
      A=2.25                                                            GISO0970
      B=-0.984375                                                       GISO0980
      CSYMB=-0.50                                                       GISO0990
      PI=3.14159265358979 3                                             GISO1000
      PI2=PI/2.0                                                        GISO1010
      ALFAP=0.0000                                                      GISO1020
      GRID=0.150                                                        GISO1030
      ENUM=5                                                            GISO1040
      ELL=GRID/7.5                                                      GISO1050
C                                                                       GISO1060
C ******************************************************************** GISO1070
C                                                                       GISO1080
C     SECTION 1.  FRONT SURFACE CONSTRUCTION                            GISO1090
C                                                                       GISO1100
C ******************************************************************** GISO1110
C                                                                       GISO1120
      BETA=DATAN(F/(F-R*DCOTAN(ALPHA)))                                 GISO1130
      DLUDP=BETA/I                                                      GISO1140
      BF=F/R                                                            GISO1150
      WRITE(4,125) BF,ALPHA,ALFAP                                       GISO1160
  125 FORMAT(' ',3F10.7)                                                GISO1170
      WRITE(6,1900)                                                     GISO1180
      IF(SHAPE .EQ. 1) WRITE(6,100) ALPHA,BETA,R,L,I,T,N1,              GISO1190
     1N3,BF,DLUDP                                                       GISO1200
C                                                                       GISO1210
C    START LENS CONSTRUCTION BY TRACING OUTERMOST RAY                   GISO1220
C                                                                       GISO1230
      FILTER=0                                                          GISO1240
      J=1                                                               GISO1250
C                                                                       GISO1260
C     FIND OUTERMOST INSIDE SURFACE POINT                               GISO1270
C                                                                       GISO1280
      UDP(J)=BETA                                                       GISO1290
      X2(J)=(BF*DSIN(UDP(J)))/(DSIN(UDP(J))+(DTAN(ALPHA))*DCOS(UDP(J))) GISO1300
      Y2(J)=(X2(J))*DTAN(ALPHA)                                         GISO1310
C                                                                       GISO1320
C     CALCULATE ALL LINE CONSTANTS                                      GISO1330
C                                                                       GISO1340
      I2P(J)=-ALPHA+DATAN((BF-X2(J))/Y2(J))                             GISO1350
      RO=DSQRT((X2(J)+CSYMB)**2+Y2(J)**2)                               GISO1360
      RZERO=RO                                                          GISO1370
      IF(CSYMB.LT.-0.6) RZERO=DSQRT(R**2+CSYMB**2)                      GISO1380
      CALL INDEX(RO,N20,A,B,RZERO)                                      GISO1390
      I2(J)=DARSIN((N3/N20)*DSIN(I2P(J)))                               GISO1400
      UP(J)=PI2-ALPHA-I2(J)                                             GISO1410
      BASE=X2(J)+CSYMB                                                  GISO1420
      IF(BASE .EQ. 0.0) CSYMB=1.0000001*CSYMB                           GISO1430
      BASE=X2(J)+CSYMB                                                  GISO1440
```

```
      THETAO=DATAN(Y2(J)/BASE)
      IF(THETAO .LT. 0.0)THETAO=PI+THETAO
      ED=T/R
      X1H=X2(J)-ED*DCOS(UP(J))
      Y1H=Y2(J)+ED*DSIN(UP(J))
      BAS1=X1H+OSYMB
      IF(BAS1 .EQ. 0.0)OSYMB=1.0000001*OSYMB
      BAS1=X1H+OSYMB
      THETAE=DATAN(Y1H/BAS1)
      IF(THETAE .LT. 0.0)THETAE=PI+THETAE
      PSIO=PI-UP(J)-THETAO
      EPSLON=+1.0
      IF(PSIO .GT. PI2.OR.PSIO .LT. 0.0) EPSLON=-1.0
      E=N20*RC*DSIN(PSIO)
         ARG=DSQRT(A**2+4*B*(E**2)/(RZERO**2))
         RSINO=DARSIN((2*(E**2)/(RO**2)-A)/ARG)
         C=PI2-RSINO
         IF(PSIO.LT.PI2) C=0.0
         TCRIT=THETAO-(EPSLON/2.)*(PI2-RSINO)
         IF(THETAE.GE.TCRIT) THETE=THETAE-C
         IF(THETAE.LT.TCRIT) THETE=THETAE
         IF(THETAE.GE.TCRIT) EPSLON=+1.0
         THETE=THETE-THETAO
C
C     FIND RADIUS CORRESPONDING TO ANGLE SPECIFIED BY EDGE THICKNESS
C     NOTE THAT NO ITERATION REQUIRED FOR FIRST RAY
C
      CALL RADYUS(A,E,ARG,RSINO,THETE,RAD,RDOTG,FILTER,EPSLON)
      IF(FILTER .EQ. 1)WRITE(6,1800)
      IF(FILTER .EQ. 1)GO TO 82
C
C     CALCULATE COORDINATES, ANGLE AND SLOPE FOR FRONT SURFACE SOLUTION
C
      X1(J)=RAD*COS(THETAE)-OSYMB
      Y1(J)=RAD*SIN(THETAE)
      CALL INDEX(RAD,N2,A,B,RZERO)
      PSI=DARSIN(E/(N2*RAD))
      IF(THETAE.LT.TCRIT) PSI=PI-PSI
      ZETA=PI-PSI-THETAE
      I1(J)=DARSIN(DSQRT(((DSIN(ZETA-U))**2)/((DCOS(ZETA-U)-(N1/N2))**2+
     1(DSIN(ZETA-U))**2)))
      I1P(J)=DARSIN((N1/N2)*DSIN(I1(J)))
      DYDXN(J)=-DTAN(I1(J)+U)
      DYDXT(J)=DCOTAN(I1(J)+U)
      IF(SHAPE .EQ. 0) GO TO 5
      WRITE(6,200) J,X1(J),Y1(J),X2(J),Y2(J),UDP(J),I2P(J),I2(J),UP(J),
     1I1P(J),I1(J),DYDXT(J)
    5 CONTINUE
```

```
C                                                                     GISO1930
C     END OF TRACE FOR 1ST RAY                                        GISO1940
C                                                                     GISO1950
C     START ITERATIVE SOLUTION SCHEME                                 GISO1960
C                                                                     GISO1970
      K=I+1                                                           GISO1980
      SURFL=0.0                                                       GISO1990
      DO 10 J=2,K                                                     GISO2000
         TRIP=0                                                       GISO2010
         L=J-1                                                        GISO2020
C                                                                     GISO2030
C      CALCULATE INNER SURFACE COORDINATES                            GISO2040
C                                                                     GISO2050
         UDP(J)=BETA-(L)*DLUDP                                        GISO2060
         X2(J)=(BF*DSIN(UDP(J)))/(DSIN(UDP(J))+(DTAN(ALPHA))*         GISO2070
     1   DCOS(UDP(J)))                                                GISO2080
         Y2(J)=(X2(J))*DTAN(ALPHA)                                    GISO2090
C                                                                     GISO2100
C       CALCULATE ALL LINE CONSTANTS                                  GISO2110
C                                                                     GISO2120
         RO=DSQRT((X2(J)+DSYMB)**2+Y2(J)**2)                          GISO2130
         IF(J .EQ. K) ROOO=RO                                         GISO2140
         I2P(J)=-ALPHA+DATAN((BF-X2(J))/Y2(J))                        GISO2150
         CALL INDEX(RO,N20,A,B,RZERO)                                 GISO2160
         I2(J)=DARSIN((N3/N20)*DSIN(I2P(J)))                          GISO2170
         UP(J)=PI2-ALPHA-I2(J)                                        GISO2180
         THETAO=DATAN(Y2(J)/(X2(J)+DSYMB))                            GISO2190
         IF(THETAO.LT.0.0) THETAO=PI+THETAO                           GISO2200
         PSIO=PI-UP(J)-THETAO                                         GISO2210
         EPSLON=1.0                                                   GISO2220
         IF(PSIO .GT. PI2.OR.PSIO.LT.0.0) EPSLON=-1.0                 GISO2230
         E=N20*RO*DSIN(PSIO)                                          GISO2240
         Y1H=((DCOTAN(I1(L)+U))*(DCOTAN(UP(J)))*(Y2(J)+               GISO2250
     1   X2(J)*DTAN(UP(J)))+Y1(L)-X1(L)*DCOTAN(I1(L)+U))/(1.0+        GISO2260
     1   (DCOTAN(I1(L)+U))*DCOTAN(UP(J)))                             GISO2270
         X1H=(DCOTAN(UP(J)))*(-Y1H+Y2(J)+X2(J)*DTAN(UP(J)))           GISO2280
         IF(B .EQ. 0.0) GO TO 8                                       GISO2290
         ARG=DSQRT(A**2+4*B*(E**2)/(RZERO**2))                        GISO2300
         RSINO=DARSIN((2*(E**2)/(RO**2)-A)/ARG)                       GISO2310
         C=PI2-RSINO                                                  GISO2320
         IF(PSIO.LT.PI2) C=0.0                                        GISO2330
         TCRIT=THETAO-(EPSLON/2.)*(PI2-RSINO)                         GISO2340
         THETAP=DATAN(Y1H/(X1H+DSYMB))                                GISO2350
         IF(THETAP.LT.0.0) THETAP=PI+THETAP                           GISO2360
         RP=DSQRT((X1H+DSYMB)**2+Y1H**2)                              GISO2370
         SL=DCOTAN(I1(L)+U)                                           GISO2380
         BE=Y1H-SL*(X1H+DSYMB)                                        GISO2390
C                                                                     GISO2400
```

```
C      IF ANGLE LARGE ENOUGH GO TO ANGLE IN TERMS OF RADIUS ITERATION
C
          IF(THETAO.GT.PI2) GO TO 106
C
C        NEWTON-RAPHSON ITERATION
C
    6     CONTINUE
          TRIP=TRIP+1
          IF(THETAP.GE.TCRIT.AND.OSYMB.LT.0.0) THETP=THETAP-C
          IF(THETAP.LT.TCRIT.AND.CSYMB.LT.0.0) THETP=THETAP
          IF(THETAP.GE.TCRIT.AND.OSYMB.LT.0.0) EPSLON=+1.0
          IF(CSYMB.GE.0.0) THETP=THETAP
          THETAH=THETP-THETAO
          CALL RADYUS(A,E,ARG,RSINO,THETAH,RAD,RDG,FILTER,EPSLON)
          IF(FILTER .EQ. 1) WRITE(6,1800)
          IF(FILTER .EQ. 1) GO TO 82
          CALL RADUS(PI,X1(L),Y1(L),I1(L),L,OSYMB,THETAP,RADD,RDGG)
          RDF=RAD-RADD
          RDT=RDG-RDGG
          FACTOR=1.7/(DABS(PSIO))**.5
          IF(CSYMB.GT.0.0) FACTOR=1.3
          THETAN=THETAP-RDF/(RDT*FACTOR)
          DIFF=DABS(THETAN-THETAP)
          EXIT=0.0000001*DABS(PSIO)
          IF(CSYMB.GT.0.0) EXIT=0.0000001
          EXIT2=300
          IF(DIFF.LT.EXIT) GO TO 7
          IF(TRIP.GT.EXIT2) GO TO 7
          THETAP=THETAN
          GO TO 6
C
C     ANGLE IN TERMS OF RADIUS ITERATION
C
  106     TRIP=TRIP+1
          RSIN=DARSIN((2*E**2/RP**2-A)/ARG)
          THETA=THETAO-EPSLON/2*(RSIN-RSINO)
          CHECK1=THETA-THETAP
          RPP=EE/(DSIN(THETA)-SL*DCOS(THETA))
          CHECK2=DABS(RPP-RP)
          THETAP=THETA
          RP=RPP
          IF(TRIP.GT.100) GO TO 107
          IF(CHECK2.LT.0.0000001) GO TO 107
          GO TO 106
  107     RAD=RPP
C
C     FIND COORDINATES, ANGLE AND SLOPE OF FRONT SURFACE FOR SOLUTION
C
```

```
    7       X1(J)=RAD*DCOS(THETAP)-OSYMB
            Y1(J)=RAD*DSIN(THETAP)
            CALL INDEX(RAD,N2,A,B,RZERO)
            EONR=E/(N2*RAD)
            IF(EONR .GT. 1.0) WRITE(6,9978)EONR
 9978       FORMAT(10X,'EONR =',F19.16)
            IF(EONR .GT. 1.0)EONR=1.0
            PSIP=DARSIN(EONR)
            IF(THETAP.LT.TCRIT.AND.PSIO.GT.0.0) PSIP=PI-PSIP
            ZETA=PI-PSIP-THETAP
            GO TO 9
C
    8    CONTINUE
         X1(J)=X1H
         Y1(J)=Y1H
         ZETA=UF(J)
C
    9    CONTINUE
         IF(Y1(J) .LE. 0.0) GO TO 11
            I1(J)=DARSIN(DSQRT((DSIN(ZETA-U))**2/((DCOS(ZETA-U)-(N1/N2))**2
     1      +(DSIN(ZETA-U))**2)))
            I1P(J)=DARSIN((N1/N2)*DSIN(I1(J)))
            DYDXN(J)=(-DTAN(I1(J)+U))
            DYDXT(J)=DCOTAN(I1(J)+U)
         SLINCR=DSQRT((X1(J)-X1(L))**2+(Y1(L)-Y1(J))**2)
         SURFL=SURFL+SLINCR
            IF(SHAPE .EQ. 0) GO TO 10
            WRITE(6,200)J,X1(J),Y1(J),X2(J),Y2(J),UDF(J),I2P(J),I2(J),
     1      UP(J),I1P(J),I1(J),DYDXT(J)
   10    CONTINUE
         GO TO 12
C
C     END OF TRACE FOR FRONT SURFACE CONSTRUCTION
C
C     OPAQUE NOSE SECTION CALCULATIONS
C
   11 CONTINUE
      K=J-1
   12 CONTINUE
      AB=(-X1(K))+(Y1(K))/DYDXT(K)
      STATNA=-AB
      GAMMA=ATAN(DYDXT(K))
      CNP=(Y1(K))/SIN(GAMMA)
      TSURFL=CNP+SURFL
      WRITE(6,300) GAMMA,STATNA,CNP,SURFL,TSURFL
C
C     WRITE OUTPUT FILE FOR PROGRAM CHECK
C
```

```
      WRITE(2,110)A,B,CSYMB,I,ALPHA,N3,F,R
      DO 120 J=1,I
  120 WRITE(2,111)X1(J),Y1(J),X2(J),Y2(J)
      WRITE(2,112)
  110 FORMAT(' ',3F11.7,I5,F10.7,3F5.3)
  111 FORMAT(' ',4F10.7)
  112 FORMAT(' ','/*')
C
C     LENS INDEX CF REFRACTION DATA -- N2(R) --
C
      BEFF=B/(RZERO**2)
      SQRE=A+BEFF*RZERO**2
      IF(CSYMB.LT.0.0) SQRE=A+BEFF*(RZERO/1.5)**2
      N2X21=DSQRT(SQRE)
      CALL INDEX(RAD,N2X1K,A,B,RZERO)
      CALL INDEX(ROOO,N2X2K,A,B,RZERO)
      PRCNTE=-((N2X2K-N2X21)/N2X2K)*100.0
      PRCNTT=-((N2X2K-N2X1K)/N2X2K)*100.0
      PRCNTO=-((N2X1K-N2X21)/N2X1K)*100.0
      WRITE(6,170C)A,BEFF,CSYMB,PRCNTE,PRCNTT,PRCNTO
C
C ******************************************************************
C
C      SECTION 2. SKEW RAY TRACE
C
C ******************************************************************
C
C
C    INITIALIZE
C
      G=1
      COUNT=C
      COUNT2=C
      RAYY(G)=C.0
      RAYZ(G)=C.0
      NTNCTY(G)=1.0
      XMAVET=C.0
      XMAVE=0.0
      IF(SQURAY .EQ. 1) WRITE(6,400) ALFAP,GRID
C
C     CCNSTUUCT GRIC
C
   2C CONTINUE
      IF(RAYZ(G) .GE. Y1(K)) GO TC 30
        CENI=(AB+X1(K))*SIN(ALFAP)
         TCRBI=CSQRT(Y1(K)**2-RAYZ(G)**2)
         BOTI=CENI-TORBI
         IF(RAYY(G) .LE. BOTI) GO TO 30
```

```
      TOPI=CENI+TORBI                                                   GIS03850
      IF(RAYY(G) .GE. TOPI) GO TO 30                                    GIS03860
        RAYZ(G)=RAYZ(G)+GRID                                            GIS03870
        GO TO 20                                                        GIS03880
   30 CONTINUE                                                          GIS03890
C  SEARCH FOR Y1(J) JUST GT RAYZ(G); RETRIEVE X1(J)                     GIS03900
      M=K/2                                                             GIS03910
      IF(Y1(M) .LT. RAYZ(G)) GO TO 32                                   GIS03920
        IF(RAYY(G) .GT. 1.15) GO TO 32                                  GIS03930
          N=K-1                                                         GIS03940
   31 CONTINUE                                                          GIS03950
      IF(Y1(N) .GT. RAYZ(G)) GO TO 36                                   GIS03960
        N=N-1                                                           GIS03970
        IF(N .EQ. 0) GO TO 34                                           GIS03980
        GO TO 31                                                        GIS03990
   32 CONTINUE                                                          GIS04000
      N=M                                                               GIS04010
   33 CONTINUE                                                          GIS04020
      IF(Y1(N) .GT. RAYZ(G)) GO TO 36                                   GIS04030
        N=N-1                                                           GIS04040
        IF(N .EQ. 0) GO TO 34                                           GIS04050
          GO TO 33                                                      GIS04060
   34 CONTINUE                                                          GIS04070
      RAYZ(G)=0.0                                                       GIS04080
      IF(RAYY(G) .LT. 0.0) GO TO 35                                     GIS04090
        RAYY(G)=RAYY(G)+GRID                                            GIS04100
        GO TO 20                                                        GIS04110
   35 CONTINUE                                                          GIS04120
      RAYY(G)=RAYY(G)-GRID                                              GIS04130
      GO TO 20                                                          GIS04140
   36 CONTINUE                                                          GIS04150
      RADIUS=DSQRT(RAYZ(G)**2+((RAYY(G)/COS(ALFAP))-(X1(N)+AB)*         GIS04160
     1TAN(ALFAP))**2)                                                   GIS04170
      IF(Y1(N) .GT. RADIUS) GO TO 40                                    GIS04180
        IF(RADIUS .GT.Y1(1)) GO TO 37                                   GIS04190
        N=N-1                                                           GIS04200
        GO TO 36                                                        GIS04210
   37 CONTINUE                                                          GIS04220
      N=N-1                                                             GIS04230
      IF(N .GT. 0) GO TO 36                                             GIS04240
      CEN=(X1(1)+AB)*SIN(ALFAP)                                         GIS04250
      TORB=Y1(1)*COS(ALFAP)                                             GIS04260
      RAYZ(G)=0.0                                                       GIS04270
      IF(RAYY(G) .GE. 0.0) GO TO 38                                     GIS04280
        RAYY(G)=RAYY(G)-GRID                                            GIS04290
        BCT=CEN-TORB                                                    GIS04300
C       CHECK FOR BOTTOM EDGE OF LENS, GO TO MIRROR IMAGE RAYS          GIS04310
        IF(RAYY(G) .LT. BCT) GO TO 60                                   GIS04320
```

```
      GO TO 20                                                          GIS04330
   38 CONTINUE                                                          GIS04340
      RAYY(G)=RAYY(G)+GRID                                              GIS04350
      TOP=CEN+TORE                                                      GIS04360
      IF (RAYY(G) .GT. TOP) GO TO 39                                    GIS04370
         GO TO 20                                                       GIS04380
   39 CONTINUE                                                          GIS04390
      RAYY(G)=-GRID                                                     GIS04400
      GO TO 20                                                          GIS04410
   40 CONTINUE                                                          GIS04420
      P=N+1                                                             GIS04430
C                                                                       GIS04440
C     CALCULATE INTERCEPT POINT ON OUTER SURFACE                        GIS04450
C                                                                       GIS04460
      AO=RAYY(G)/COS(ALFAP)                                             GIS04470
      BO=TAN(ALFAP)                                                     GIS04480
      CO=(Y1(N)-Y1(P))/(X1(N)-X1(P))                                    GIS04490
      DO=X1(P)                                                          GIS04500
      EO=Y1(P)                                                          GIS04510
      PAR1=(AC*BO+CO*EO-(CO**2)*DO-AB*(BC**2))                          GIS04520
      PAR2=(EC**2-CO**2)                                                GIS04530
      PAR3=(AC**2+AB*(BO**2)*AB-2.0*AO*BC*AB+                           GIS04540
     1RAYZ(G)**2-(EO-CO*DO)**2)                                         GIS04550
      XO=(PAR1/PAR2)+SQRT((PAR1/PAR2)**2-PAR3/PAR2)                     GIS04560
      IF((PAR1/PAR2).GT.X1(1))XO=XO-2.0*SQRT((PAR1/PAR2)**2-PAR3/PAR2)  GIS04570
      YO=(RAYY(G)/COS(ALFAP))-(XO+AB)*TAN(ALFAP)                        GIS04580
      ZO=RAYZ(G)                                                        GIS04590
C                                                                       GIS04600
C     CALCULATE THE DIRECTION COSINES OF OUTSIDE SURFACE NORMAL         GIS04610
C                                                                       GIS04620
      DELTAX=X1(N)-X1(P)                                                GIS04630
      DELTAY=Y1(N)-Y1(P)                                                GIS04640
      SP=DSQRT((RADIUS-Y1(P))**2+(XO-X1(P))**2)                         GIS04650
      ST=SQRT(DELTAX**2+DELTAY**2)                                      GIS04660
      RATIO=SP/ST                                                       GIS04670
      DYDXNP=RATIO*(DYDXN(N)-DYDXN(P))+DYDXN(P)                         GIS04680
      NPI=1.0/DYDXNP                                                    GIS04690
      IF (YO .EQ. 0.0) GO TO 41                                         GIS04700
         ZOYO=(ZO/YO)                                                   GIS04710
         IF(ZOYO .GE. 8235000) GO TO 41                                 GIS04720
         IF(ZOYO .LE. -8235000) GO TO 42                                GIS04730
           NPJ=DCOS(DATAN(ZO/YO))                                       GIS04740
           NPK=DSIN(DATAN(ZO/YO))                                       GIS04750
           GO TO 43                                                     GIS04760
   41 CONTINUE                                                          GIS04770
      NPJ=0.0                                                           GIS04780
      NPK=1.0                                                           GIS04790
      GO TO 43                                                          GIS04800
```

```
   42 CONTINUE                                                          GIS04810
      NPJ=0.0                                                           GIS04820
      NPK=-1.0                                                          GIS04830
   43 CONTINUE                                                          GIS04840
C                                                                       GIS04850
C     DIRECTION COSINES OF EXTERNAL RAY                                 GIS04860
C                                                                       GIS04870
      CK=COS(ALFAP)                                                     GIS04880
      CL=-SIN(ALFAP)                                                    GIS04890
C                                                                       GIS04900
C     DIRECTION COSINES OF OUTSIDE SURFACE NORMAL                       GIS04910
C                                                                       GIS04920
      LK=NPI/(SQRT(NPI**2+NPJ**2+NPK**2))                               GIS04930
      LL=(NPJ*LK)/NPI                                                   GIS04940
      LM=(NPK*LK)/NPI                                                   GIS04950
      IF(YO .LT. 0.0) GO TO 44                                          GIS04960
        GO TO 45                                                        GIS04970
   44 CONTINUE                                                          GIS04980
      LL=-LL                                                            GIS04990
      LM=-LM                                                            GIS05000
   45 CONTINUE                                                          GIS05010
      THETA=DARCOS(CK*LK+CL*LL)                                         GIS05020
C                                                                       GIS05030
C     ANGLES OF INCIDENCE AND REFRACTION OUTSIDE SURFACE                 GIS05040
C                                                                       GIS05050
      PHI=PI-THETA                                                      GIS05060
      RO=DSQRT((XL+OSYMB)**2+YO**2+ZO**2)                               GIS05070
      CALL INDEX(RO,N2O,A,B,RZERO)                                      GIS05080
      ART=(N1/N2O)*DSIN(PHI)                                            GIS05090
      IF(ART.GT.1.0) ART=1.0                                            GIS05100
      IF(ART.LT.-1.0) ART=-1.0                                          GIS05110
      PHIP=DARSIN(ART)                                                  GIS05120
C                                                                       GIS05130
C     CALCULATION OF TRANSMITED INTENSITY AT OUTSIDE SURFACE            GIS05140
C                                                                       GIS05150
      FACE=1                                                            GIS05160
      CALL XMIT(PHI,N2O,XMTNC,FACE,XM,N1,N3)                            GIS05170
      IF(FACE .EQ. 3) GO TO 53                                          GIS05180
      NUM=DCOS(PHIP)-(N1/N2O)*DCOS(PHI)                                 GIS05190
C                                                                       GIS05200
C     INITIAL DIRECTION COSINES OF INTERNAL REFRACTED RAY               GIS05210
C                                                                       GIS05220
      CKP=(N1/N2O)*CK-NUM*LK                                            GIS05230
      CLP=(N1/N2O)*CL-NUM*LL                                            GIS05240
      CMP=-NUM*LM                                                       GIS05250
      CKKP=CKP                                                          GIS05260
      CLLP=CLP                                                          GIS05270
      CMMP=CMP                                                          GIS05280
```

```
C
C     INTERNAL RAY LENGTH AND INSIDE SURFACE INTERCEPT
C
      ROX=(XO+OSYMB)/RO
      ROY=YO/RO
      ROZ=ZO/RO
      NPOX=ROY*CMP-ROZ*CLP
      NPOY=ROZ*CKP-ROX*CMP
      NPOZ=ROX*CLP-ROY*CKP
      NPO=DSQRT(NPOX**2+NPOY**2+NPOZ**2)
      NPOX=NPOX/NPO
      NPOY=NPOY/NPO
      NPOZ=NPOZ/NPO
C
C  CALCULATE PSIO AND THEN DETERMINE ITS SIGN
C
      PSIO=DARCOS(ROX*CKP+ROY*CLP+ROZ*CMP)
      IF(NPOZ.EQ.0.0) XX=OSYMB*ROZ/(ROX*NPOY)*(NPOY*ROY/ROZ+NPOZ)
      IF(NPOZ.EQ.0.0) GO TO 121
      XX=OSYMB*ROY/(ROX*NPOZ)*(NPOZ*ROZ/ROY+NPOY)
  121 CONTINUE
      CCK=OSYMB/DSQRT(OSYMB**2+XX**2)
      CCL=0.0
      CCM=XX/DSQRT(OSYMB**2+XX**2)
      IF(NPOZ.EQ.0.0) CCL=XX/DSQRT(OSYMB**2+XX**2)
      IF(NPOZ.EQ.0.0) CCM=0.0
      THETA1=PI-DARCOS(CCK*CKP+CCL*CLP+CCM*CMP)
      THETA2=DARCOS(CCK*ROX+CCL*ROY+CCM*ROZ)
      FLUG=0
      IF(THETA1.GT.THETA2.AND.OSYMB.LT.0.0) FLUG=1
      IF(FLUG.EQ.1) PSIO=-PSIO
C
C  CALCULATE LINE CONSTANTS
C
      EPSLON=+1.0
      IF(PSIO.LT.PI.AND.PSIO.GT.PI2) EPSLON=-1.0
      E=N20*RO*DSIN(PSIO)
      ARG=DSQRT(A**2+4*B*E**2/RZERO**2)
      RSINO=DARSIN((2*E**2/RO**2-A)/ARG)
      C=PI2-RSINO
      IF(PSIO.LT.PI2) C=0.0
      TCRIT=-EPSLON/2*(PI2-RSINO)
      NUM1=(CLP*YO+CMP*ZO-CKP*XO*(DTAN(ALPHA)**2))
      NUM2=(CLP**2+CMP**2-(CKP**2)*(DTAN(ALPHA)**2))
      NUM3=(YO**2+ZO**2-(XO**2)*(DTAN(ALPHA)**2))
      NUM4=NUM1**2-NUM2*NUM3
C
C     CHECK FOR INTERCEPT WITH INSIDE SURFACE
```

```
C
      IF(NUM4 .LT. 0.0) GO TO 50
C
C     CALCULATE FIRST GUESS POINT
C
        D2=-NUM1/NUM2-DSQRT(NUM4)/NUM2
          XIH=D2*CKP+XO
          YIH=D2*CLP+YO
          ZIH=D2*CMP+ZO
          IF(ZIH .LT. 0.0) GO TO 50
          OSYMPH=DSQRT((CSYMB+XIH)**2+YIH**2+ZIH**2)
          OPHX=(CSYMB+XIH)/CSYMPH
          OPHY=YIH/OSYMPH
          OPHZ=ZIH/OSYMPH
          THETAP=DARCOS(ROX*OPHX+ROY*OPHY+ROZ*OPHZ)
          RP=CSYMPH
          IF(E.EQ.0.0) GO TO 124
          SIGN=-1.0
          IF(CSYMB .GT. 0.0) SIGN=+1.0
          IF(PSIO.LT.0.0) THETAP=-THETAP
C
C  INTIALIZE SURF ROUTINE TO FIND CORRECT VALUE OF SIGN
C
      CALL SURF(THETAP,RADD,RDOTS,SIGN,FILTER,NPOX,NPOY,NPOZ,PSIO,
     1XO,YO,ZO,OSYMB,RO,CKKP,CLLP,CMMP,ALPHA)
          IF(RADD .LT. 0.0) SIGN=-SIGN
          THETP=PI-THETA+THETAP
C
C   CALL NEWTON-RAPHSON ITERATION IF ANGLE LARGE ENOUGH
C
          IF(DABS(PSIO).GT.3.0.AND.OSYMB.LT.0.0)
     1    CALL INTER2(RP,THETAP,RAD,THTAP,A,E,ARG,RSIND,EPSLON,SIGN,
     1    NPOX,NPOY,NPOZ,PSIO,XO,YO,ZO,OSYMB,RO,CKKP,CLLP,CMMP,
     1    ALPHA,THETA,FILTER,FLUG)
          IF(DABS(PSIO).GT.3.0.AND.OSYMB.LT.0.0) GO TO 118
C
C   CALL ANGLE IN TERMS OF RADIUS ROUTINE IN ANGLE SMALL ENOUGH
C
          CALL INTER1(SIGN,THETAP,RAD,THTAP,FILTER,A,E,ARG,RSIND,
     1EPSLON,C,TCRIT,NPOX,NPOY,NPOZ,PSIO,XO,YO,ZO,RO,CKKP,CLLP,CMMP,
     1ALPHA,CSYMB,FLUG)
  118     CONTINUE
C
C   CHECK FOR FAILED RAYS
C
          IF(FILTER .EQ. 2 .OR. FILTER .EQ. 1) GO TO 51
          IF(FILTER .EQ. 3) GO TO 50
          IF(FILTER .EQ. 4) GO TO 50
```

```
C
          GO TO 123
 124      THTAP=THETAP
          RAD=RP
C
C   CALCULATE ALL REQUIRED PARAMETERS AT SOLUTION POINT
C
 123      BETAO=DSIN(THTAP)/DSIN(PSIO)
          ALFAO=DCOS(THTAP)-BETAO*DCOS(PSIO)
          XI=RAD*(ALFAO*((XO+DSYMB)/RO)+BETAO*CKKP)-DSYMB
          YI=RAD*(ALFAO*(YO/RO)+BETAO*CLLP)
          ZI=RAD*(ALFAO*(ZO/RO)+BETAO*CMMP)
          CALL INDEX(RAD,N2,A,B,RZERO)
          EONR=DABS(E/(N2*RAD))
          IF(EONR .GT. 1.0) WRITE(6,9978)EONR
          IF(EONR .GT. 1.0) EONR=1.0
          PSIP=DARSIN(EONR)
          IF(CSYMB.LT.0.0.AND.TCRIT.GT.THTAP) PSIP=PI-PSIP
C
C   CALCULATE REFRACTION OF RAY AT INNER SURFACE
C
          A1=(XI+CSYMB)/RAD
          B1=YI/RAD
          C1=ZI/RAD
          CALL DIRECT(CKP,CLP,CMP,DIR,A1,B1,C1,NPOX,NPOY,NPOZ,PSIP)
        IF(XI .LT. 0.0 .OR. XI .GT. X2(1)) GO TO 51
          NUM5=DSQRT((XI**2)*(DTAN(ALPHA)**4)+YI**2+ZI**2)
C
C         DIRECTION COSINES OF INSIDE NORMAL
C
          LKP=-XI*(DTAN(ALPHA)**2)/NUM5
          LLP=YI/NUM5
          LMP=ZI/NUM5
          KKLLMM=CKP*LKP+CLP*LLP+CMP*LMP
          IF(KKLLMM .GE. 1.0)THETAI=0.0
          IF(KKLLMM .LE. -1.0)THETAI=PI
          IF((DABS(KKLLMM)) .GE. 1.0) GO TO 457
          THETAI=DARCOS((CKP*LKP)+(CLP*LLP)+(CMP*LMP))
  457     CONTINUE
C
C         ANGLES OF INCIDENCE AND TOTAL INTERNAL REFLECTION
C         AT THE INSIDE SURFACE:
C
          PHII=PI-THETAI
          IF(FLLG.EQ.1) PHII=THETAI
          TIRA=DARSIN(N3/N2)
C
C         CHECK FOR TOTAL INTERNAL REFLECTION:
```

```
C
      IF(PHII .GE. TIRA) GO TO 52
         PHIIP=DARSIN((N2/N3)*DSIN(PHII))
C
C        CALCULATION OF TRANSMITTED INTENSITY AT INSIDE SURFACE
C
         FACE=2
         CALL XMIT(PHII,N2,XMTNCP,FACE,XM,N1,N3)
         NTNCTY(G)=XMTNC*XMTNCP
         XMAVET=XMAVET+NTNCTY(G)
         XMAVE=XMAVET/(G*1.0)
         NUM8=DCOS(PHIIP)-(N2/N3)*DCOS(PHII)
C
C   CALCULATE DIRECTION COSINES OF INSIDE EXTERNAL REFRACTED RAY
C   CORRECTING FOR RAYS PAST REGION OF NEAR-RADIAL LINES
C
      IF(FLUG.EQ.1) CKP=-CKP
      IF(FLUG.EQ.1) CLP=-CLP
      IF(FLUG.EQ.1) CMP=-CMP
         CKPP=(N2/N3)*CKP-NUM8*LKP
         CLPP=(N2/N3)*CLP-NUM8*LLP
         CMPP=(N2/N3)*CMP-NUM8*LMP
C
C   WRITE OUTPUT FOR PROGRAM DETPLOT
C
         WRITE(4,126) XI,YI,ZI,CKPP,CLPP,CMPP
 126     FORMAT(' ',6F10.7)
C
C        LENGTH OF INSIDE EXTERNAL REFRACTED RAY
C
         D3=(BF-XI)/CKPP
         NUM6=((XO+AB)*CK+YO*CL)
         NUM7=((XO+AB)*(-CL)+YO*CK)
C
C        LENGTH OF OUTSIDE INCIDENT RAY
C
         D1=DSQRT((XO-NUM6)**2+(YO-NUM7)**2)
C
C        TOTAL OPTICAL PATHLENGTH:
C
         OPL(G)=N1*D1+N2*D2+N3*D3
C
C        INTERSECTION WITH THE IMAGE PLANE:
C
         YIM(G)=((BF-XI)/CKPP)*CLPP+YI
         ZIM(G)=((BF-XI)/CKPP)*CMPP+ZI
         DIM(G)=DSQRT(YIM(G)**2+ZIM(G)**2)
         IF(DABS(YIM(G)).GT.1.0.OR.DABS(ZIM(G)).GT.1.0.OR.DIM(G).GT.
```

```
1 1.0)    GC TO 55
          IF(CMPP .LT. 0.000000000001 .AND. CMPP .GT. -0.000000000001)
   1        GO TO 46
            XDIAPT(G)=(-ZIM(G)/CMPP)*CKFP+AE
            YDIAPT(G)=(-ZIM(G)/CMPP)*CLFP+YIM(G)
            GC TO 47
   46 CONTINUE
          XDIAPT(G)=99999.
          YDIAPT(G)=99999.
   47 CONTINUE
          IF(SQURAY .EQ. 1) WRITE(6,500)G,RAYY(G),RAYZ(G),XG,YG,XI,
   1      YI,ZI,OPL(G),YIM(G),ZIM(G),NTNCTY(G),XDIAPT(G),YDIAPT(G)
          FLAG(G)=0
   49 CONTINUE
      H=G
      G=H+1
      NTNCTY(G)=1.0
      RAYY(G)=RAYY(H)
      RAYZ(G)=RAYZ(H)+GRID
      IF(FLAG(H) .EQ. 0) COUNT=CCUNT+1
      GO TO 3C
C
C     END OF RAY TRACE
C
C     FLAG ALL FAILED RAYS
C
C     NO INTERCEPT WITH 2ND SURFACE
   50 NTNCTY(G)=0.0
      FLAG(G)=1
      IF(SQURAY .EQ. 1) WRITE(6,600)G,RAYY(G),RAYZ(G)
      GO TO 49
C     INTERCEPT OUTSIDE BOUNDARY OF SECOND SURFACE
   51 NTNCTY(G)=0.0
      FLAG(G)=2
      IF(SQURAY .EQ. 1) WRITE(6,700)G,RAYY(G),RAYZ(G)
      GO TO 49
C     TOTAL INTERNAL RELECTION AT 2ND SURFACE
   52 NTNCTY(G)=0.0
      FLAG(G)=3
      IF(SQURAY .EQ. 1) WRITE(6,800)G,RAYY(G),RAYZ(G)
      GO TO 49
C     TOTAL EXTERNAL REFLECTICN
   53 NTNCTY(G)=0.0
      FLAG(G)=4
      IF(SQURAY .EQ. 1) WRITE(6,802) G,RAYY(G),RAYZ(G)
      GO TO 49
C     RAY OUTSIDE LENS DIAMETER
   55 NTNCTY(G)=0.0
```

```
      COUNT2=COUNT2+1                                                   GIS07690
      FLAG(G)=5                                                         GIS07700
      IF(SQURAY .EQ. 1) WRITE(6,804) G,RAYY(G),RAYZ(G)                  GIS07710
      GO TO 49                                                          GIS07720
C     FAILED INTER2 SUBROUTINE                                          GIS07730
 56   NTNCTY(G)=0.0                                                     GIS07740
      IF(SQURAY .EQ. 1) WRITE(6,806) G,RAYY(G),RAYZ(G)                  GIS07750
      GO TO 49                                                          GIS07760
C                                                                       GIS07770
C     MIRROR IMAGE RAY MATCHING FOLLOWS                                 GIS07780
C                                                                       GIS07790
   60 CONTINUE                                                          GIS07800
      IF(SQURAY .EQ.1) WRITE(6,401)                                     GIS07810
      Q=G-1                                                             GIS07820
      S=G-Q                                                             GIS07830
   61 CONTINUE                                                          GIS07840
      IF(RAYZ(S) .EQ. 0.0) S=S+1                                        GIS07850
         IF(S .GT. Q) GO TO 70                                          GIS07860
            FLAG(G)=FLAG(S)                                             GIS07870
            IF(FLAG(G) .EQ. 0) COUNT=COUNT+1                            GIS07880
            RAYY(G)=RAYY(S)                                             GIS07890
            RAYZ(G)=-RAYZ(S)                                            GIS07900
            IF(FLAG(S) .GE. 1) GO TO 63                                 GIS07910
               XDIAPT(G)=XDIAPT(S)                                      GIS07920
               YDIAPT(G)=YDIAPT(S)                                      GIS07930
               YIM(G)=YIM(S)                                            GIS07940
               ZIM(G)=-ZIM(S)                                           GIS07950
               OPL(G)=OPL(S)                                            GIS07960
               NTNCTY(G)=NTNCTY(S)                                      GIS07970
               IF(SQURAY .EQ. 1) WRITE(6,501)G,S,RAYY(G),RAYZ(G),OPL(G),GIS07980
     1         YIM(G),ZIM(G),NTNCTY(G)                                  GIS07990
   62 CONTINUE                                                          GIS08000
      S=S+1                                                             GIS08010
      G=G+1                                                             GIS08020
      GO TO 61                                                          GIS08030
C                                                                       GIS08040
C    FLAG ALL MIRROR IMAGE RAYS                                         GIS08050
C                                                                       GIS08060
C     NO INTERCEPT WITH 2ND SURFACE                                     GIS08070
 63   IF(FLAG(S) .GT. 1) GO TO 64                                       GIS08080
         IF(SQURAY .EQ. 1 ) WRITE(6,601)G,S,RAYY(G),RAYZ(G)             GIS08090
         GO TO 62                                                       GIS08100
C     INTERCEPT OUTSIDE BOUNDARY OF 2ND SURFACE                         GIS08110
 64   IF(FLAG(S) .GT. 2) GO TO 65                                       GIS08120
         IF(SQURAY .EQ. 1) WRITE(6,701)G,S,RAYY(G),RAYZ(G)              GIS08130
         GO TO 62                                                       GIS08140
C     TOTAL EXTERNAL REFLECTION                                         GIS08150
 65   IF(FLAG(S) .GT. 3) GO TO 66                                       GIS08160
```

```
      IF(SQURAY .EQ. 1) WRITE(6,801)G,S,RAYY(G),RAYZ(G)
      GO TO 62
C     CONTINUE
   66 IF(FLAG(S) .GT. 4) GO TO 67
      IF(SQURAY .EQ. 1) WRITE(6,803) G,S,RAYY(G),RAYZ(G)
      GO TO 62
C     RAY OUTSIDE LENS DIAMETER
   67 IF(SQURAY .EQ. 1) WRITE(6,805) G,S,RAYY(G),RAYZ(G)
      COUNT2=COUNT2+1
      GO TO 62
   70 CONTINUE
      RAYS=G-1
      DRAYS=RAYS-C
      WRITE(6,1350) OSYMB,RZERO
      WRITE(6,900)RAYS,DRAYS,COUNT2,COUNT
      XI=1.0
      YI=1.0
      ZI=1.0
      WRITE(4,127) XI,YI,ZI,CKPP,CLPP,CMPP
  127 FORMAT(' ',6F10.7)
      WRITE(4,128)
  128 FORMAT(' ','/*')
C
C     GENERATE OBJECT PLANE ELLIPSES:
C
      IF(ELLIPS .EQ. 1) WRITE(6,1000)ENUM
      NUMBR=1
      NUME=1
   71 CONTINUE
      COORD=1
      ELZ(NUMB,COORD)=-Y1(NUMBR)
   72 CONTINUE
      ELY(NUMB,COORD)=(X1(NUMBR)+AB)*SIN(ALFAP)+DSQRT((Y1(NUMBR))**2-
     1(ELZ(NUMB,COORD))**2)*COS(ALFAP)
      ELZZ=ELZ(NUMB,COORD)
      IF(ELLIPS .EQ. 1) WRITE(6,1100)NUMBR,COORD,ELY(NUMB,COORD),
     1ELZ(NUMB,COORD)
      COORD=COORD+1
      ELZ(NUMB,COORD)=ELZZ+ELL
      IF(ELZ(NUMB,COORD) .GE. Y1(NUMBR)) GO TO 73
        GO TO 72
   73 CONTINUE
      ELZZ=ELZ(NUMB,COORD)
   74 CONTINUE
      ELZ(NUMB,COORD)=ELZZ-ELL
      IF(ELZ(NUMB,COORD) .LE. -Y1(NUMBR)) GO TO 75
      ELY(NUMB,COORD)=(X1(NUMBR)+AB)*SIN(ALFAP)-DSQRT((Y1(NUMBR))**2-
     1(ELZ(NUMB,COORD))**2)*COS(ALFAP)
```

```
      ELZZ=ELZ(NUMB,COORD)
      IF(ELLIPS .EQ. 1) WRITE(6,1100)NUMBR,COORD,ELY(NUMB,COORD),
     1ELZ(NUMB,COORD)
      COOR(NUMB)=COORD
      COORD=COORD+1
      GO TO 74
   75 CONTINUE
      NUMBR=NUMBR+(K-1)/ELLNUM
      NUMB=NUMB+1
      IF(NUMBR .GT. K) GO TO 76
      GO TO 71
   76 CONTINUE
      IF(ELLIPS .EQ. 1) WRITE(6,1200)
      NUMET=NUMB-1
C
C     IMAGE PLANE SPOT DIAGRAM STATISTICAL ANALYSIS:
C
      SUM1=0.0
      DO 77 G=1,RAYS
        IF(FLAG(G) .GT. 0) GO TO 77
          SUM1=YIM(G)+SUM1
   77 CONTINUE
C     REFERENCE EACH ITERATION TO INCIDENT ANGLE & THICKNESS BY:
      IANGLE=ALFAP*10+1
      THICK=(T/R)*100
C     CENTROID OF SPOT:
      YCENTR(IANGLE,THICK)=SUM1/COUNT
C     STANDARD DEVIATIONS:
      SUM2=0.0
      SUM3=0.0
      DO 78 G=1,RAYS
        IF(FLAG(G) .GT. 0) GO TO 78
        SUM2=ZIM(G)**2+SUM2
        SUM3=(YCENTR(IANGLE,THICK)-YIM(G))**2+SUM3
   78 CONTINUE
      SIGMAZ(IANGLE,THICK)=SUM2/COUNT
      SIGMAY(IANGLE,THICK)=SUM3/COUNT
C     ROOT MEAN SQUARE SPOT SIZE:
      RMSRAD(IANGLE,THICK)=SQRT(SIGMAZ(IANGLE,THICK)+SIGMAY(IANGLE,
     1THICK))
      WRITE(6,1300) T,U,ALFAP,R,YCENTR(IANGLE,THICK),
     1SIGMAY(IANGLE,THICK),SIGMAZ(IANGLE,THICK),RMSRAD(IANGLE,THICK)
C
C     SPOT DIAGRAM ENERGY DENSITY VS. RADIUS FROM CENTROID:
C
      DO 79 G=1,RAYS
        IF(FLAG(G) .GT. 0) GO TO 79
        SDRAD(G)=DSQRT((YCENTR(IANGLE,THICK)-YIM(G))**2+ZIM(G)**2)
```

```
   79 CONTINUE
      GG=1
      ROC(GG)=0.005
      SUM4(GG)=0.0
   80 CONTINUE
      DO 81  G=1,RAYS
        IF(FLAG(G) .GT. 0) GO TO 81
        IF(SFRAC(G) .LE. ROC(GG)) SUM4(GG)=1+SUM4(GG)
   81 CONTINUE
      FRACTN(GG)=SUM4(GG)/RAYS
C     WRITE(6,1400) GG,ROC(GG),FRACTN(GG)
      HH=GG
      GG=GG+1
      ROC(GG)=ROC(HH)+0.005
      SUM4(GG)=0.0
      IF(ROC(GG) .GT. 0.5) GO TO 82
      GO TO 80
   82 CONTINUE
      GGT=GG-1
      WRITE(6,1500)
      WRITE(6,1600)
C
C***********************************************************************
C  END OF PROGRAM
C***********************************************************************
C
C     WRITE OUTPUT FOR PLOTTING PURPOSES
C
C
      WRITE(1,83)F,ALPHA,U,I,R,T,N1,N2,N3,ALFAP,GRID,STATNA,GAMMA,RAYS,
     1COUNT,K,GGT,IANGLE,THICK,USYMB,TSURFL,BEFF,PRCNTO,PRCNTE,PRCNTT,
     1XMAVE,A,E
   83 FORMAT(' ',3F9.7,I4,4F9.7/' ',3F9.7,F9.5,F9.7/' ',6I4,4F11.7/1X,
     15F11.7)
      DO 85  J=1,K
         WRITE(1,84)X1(J),Y1(J),X2(J),Y2(J)
   84    FORMAT(' ',4F10.7)
   85 CONTINUE
      DO 88  G=1,RAYS
         WRITE(1,86)RAYY(G),RAYZ(G),FLAG(G)
   86    FORMAT(' ',2F10.7,3X,I1)
         IF(FLAG(G) .GT. 0) GO TO 88
           WRITE(1,87)YIM(G),ZIM(G)
   87      FORMAT(' ',2F10.7)
   88 CONTINUE
      DO 90  GG=1,GGT
         WRITE(1,89)ROC(GG),FRACTN(GG)
   89    FORMAT(' ',F10.7,2X,F10.7)
```

```
90 CONTINUE
   WRITE(1,91)YCENTR(IANGLE,THICK),SIGMAY(IANGLE,THICK),
  1SIGMAZ(IANGLE,THICK),RMSRAD(IANGLE,THICK),NUMBT
91 FORMAT(' ',4F10.7,I4)
   DO 96 NUMB=1,NUMBT
      WRITE(1,92)COOR(NUMB)
92    FORMAT(' ',I3)
      COORT=COOR(NUMB)
      DO 95 COORD=1,COORT
         WRITE(1,94)ELY(NUMB,COORD),ELZ(NUMB,COORD)
94       FORMAT(' ',F11.8,3X,F11.8)
95    CONTINUE
96 CONTINUE
   WRITE(1,97)
97 FORMAT(' ','/*')
C
C  FORMAT STATMENTS
C
100 FORMAT('1',' LENS PARAMETERS:'///5X,'ALPHA = ',F9.7,2X,
   1'BETA = ',F9.7,2X,'RADIUS = ',F7.5//5X,'INCIDENT ANGLE = ',
   1F9.6,2X,'ITERATIONS = ',I5//5X,'EDGE THICKNESS = ',F9.7,2X,
   1'INDICES OF REFRACTION: N1 = ',F7.5//5X,'N3 = ',
   1F7.5,2X,'FOCAL LENGTH FROM STATION ZERO = ',F7.5,2X,
   1'DELTA UCP = ',F7.5///
   11X,'  J        X1          Y1           X2          Y2         UDP        I2P
   1         I2'//25X,'LP           I1P            I1        DYDXT'//)
200 FORMAT(1X,I4,7F10.7/20X,4F10.7/)
300 FORMAT('C'/1X,'NOSE SECTION DATA:'//2X,'NOSE HALF ANGLE = ',
   1F7.5,2X,'STATION A = ',F9.5/2X,'OPAQUE SURFACE LENGTH = ',F7.5/2X,
   1'TRANSPARENT SURFACE LENGTH =',F7.5/2X,'TOTAL LENS SURFACE LENGTH
   1=',F7.5///)
400 FORMAT('1','SKEW RAY TRACE PARAMETERS:'///5X,'ALFAP= ',F9.7,2X,
   1'GRID= ',F9.7,2X,' SEE LENS PARAMETERS ABOVE.'///2X,'RAY         RAYY
   1    RAYZ          X0         Y0          XI          YI          ZI'//14X,'O
   1PL          YIM         ZIM        NTNCTY          XDIAPT          YDIAPT'///)
401 FORMAT('1'/////'MIRROR IMAGE SKEW RAYS FOLLOW'///1X,'IMAGE'/2X,'RA
   1Y   RAY    RAYY          RAYZ          CPL           YIM         ZIM        NTNCT
   1Y'//)
500 FORMAT(1X,I4,7F10.7/8X,F13.9,3F10.7,2E13.3/)
501 FORMAT(I4,1X,I4,2F10.7,F13.9,3F10.7/)
600 FORMAT(1X,I4,2F10.7,' NO INTERCEPT WITH 2ND SURFACE'/)
601 FORMAT(I4,1X,I4,2F10.7,' NO INTERCEPT WITH 2ND SURFACE'/)
700 FORMAT(1X,I4,2F10.7,' INTERCEPT OUTSIDE BOUNDARY OF 2ND SURFACE'/)
701 FORMAT(I4,1X,I4,2F10.7,' INTERCEPT OUTSIDE OF BOUNDARY OF 2ND SURF
   1ACE'/)
800 FORMAT(1X,I4,2F10.7,2X,'TOTAL INTERNAL REFLECTION AT 2ND SURFACE'/
   1)
801 FORMAT(I4,1X,I4,2F10.7,'  TOTAL INTERNAL REFLECTION'/)
```

```
802  FORMAT(1X,I4,2F10.7,2X,'   TOTAL EXTERNAL REFLECTION'/)
803  FORMAT(I4,1X,I4,2F10.7,'   TOTAL EXTERNAL REFLECTION'/)
804  FORMAT(1X,I4,2F10.7,2X,'   RAY OUTSIDE LENS DIAMETER'/)
805  FORMAT(I4,1X,I4,2F10.7,'   RAY OUTSIDE LENS DIAMETER'/)
806  FORMAT(1X,I4,2F10.7,' FAILED RAY IN INTER2 ARSIN'/)
900  FORMAT(1X,'END OF SKEW RAY TRACE.'//3X,'TOTAL NUMBER OF RAYS TRACE
    1D = ',I4/3X,'TOTAL NUMBER OF MIRROR IMAGE RAYS = ',I4/3X,'TOTAL NU
    1MBER OF RAYS OUTSIDE LENS DIAMETER = 'I4/3X,'TOTAL NUMBER OF RAYS
    1STRIKING IMAGE PLANE = ',I4)
1000 FORMAT('1','TWO DIMENSIONAL PROJECTION OF 3-DIM LENS'///3X,'GENERA
    1TED ELLIPSES ARE REFERENCED TO THE CORRESPONDING JTH COORD'/3X,'IN
    1THE MERIDIAN PLANE AS COMPUTED IN THE ABOVE'/3X,'GLM LENS SHAPE AL
    1GORITHM. INSIDE/OUTSIDE ELLIPSES DEFINE THE '/3X,'LIMITS OF THE TR
    1ANSPARENT/OPAQUE PARTS.'/3X,'THE NUMBER OF ELLIPSES =  ',I3//1X,'
    1 J    COORD      ELY          EL2'//)
1100 FORMAT(1X,I4,3X,I4,2F10.7)
1200 FORMAT(1X/'CALCULATION OF ELLIPSE COORDINATES COMPLETE.')
1300 FORMAT(1X,'IMAGE PLANE SPOT DIAGRAM ANALYSIS:'/1X,'THICKNESS
    1 =',F9.7,'  U =',F9.7,'  ALFAP =',F9.7,'  R =',F9.7/1X,'CENTROID:
    1 ZCENTR = 0.0,   YCENTR = ',F10.7/1X,'STANDARD DEVIATIONS:  SIGMAY
    1 = ',F10.7,2X,'SIGMAZ = ',F10.7/1X,'RMS SPOT SIZE: RMSRAD = ',
    1F10.7)
1350 FORMAT(1X,//1X,'CSYMB =',1X,F6.3,/1X,'RZERO=',1X,F6.3)
1400 FORMAT(I4,2F11.7)
1500 FORMAT(1X/'SPOT DIAGRAM CALCULATIONS COMPLETE.'////)
1600 FORMAT(1X/'END OF PROGRAM'////)
1700 FORMAT(1X,'LENS INDEX OF REFRACTION DATA'/2X,'THE INDEX GRADIENT I
    1S: N2(R) = SQRT((',F4.2,')+(',F6.3,')*R**2)'/2X,'WITH THE CENTER O
    1F SYMMETRY LOCATED AT ',F7.3/2X,'(MINUS IMPLIES THE OSYM IS ON THE
    1 IMAGE SIDE).'/2X,'PERCENT CHANGE OF THE LENS INDEX GRADIENT:'/
    14X,'INSIDE SURFACE FROM CENTER TO EDGE :',F10.5/4X,'INSIDE SURFACE
    1 TO OUTSIDE SURFACE AT THICKEST POINT :',F10.5/4X,'OUTSIDE SURFACE
    1 FROM CENTER TO EDGE :',F10.5////)
1800 FORMAT(1X,'IMAGINARY RADIUS DUE TO IMPROPER A TO B RATIO IN N2(R)
    1FOR THIS RAY')
1900 FORMAT(1X,'*****************************************************
    1**************'/1X,'*',68X,'*'/1X,'*',25X,'GISL TABULAR OUTPUT',2
    14X,'*'/1X,'*',68X,'*'/1X,'*****************************************
    1******************************'///)
     STOP
     END
C
C
C     SUBROUTINE INDEX CALCULATES THE INDEX OF REFRACTION AS A FUNCTION
C      OF RADIUS FROM THE CENTER OF SYMMETRY. THE VALUES OF THE INDEX
C      CONSTANTS A AND B ARE USER INPUTS.
C
      SUBROUTINE INDEX(R,N,A,B,RZERO)
```

```
      DOUBLE PRECISION R,N,A,B,RZERO
      N=DSQRT(A+B*(R/RZERO)**2)
      RETURN
      END
C
C
C    SUBROUTINE TRANSMIT CALCULATES THE TRANSMITTED INTENSITY OF
C     EACH RAY AT BOTH THE OUTSIDE AND INSIDE SURFACES USING THE
C     FRESNEL EQUATIONS.
C
      SUBROUTINE XMIT(PHI,N2,XM1,FACE,XM,N1,N3)
      REAL PHI,N3,XM1,NTI,N1,N2,SQRT,COS
      INTEGER FACE
      NTI=N2/N1
      IF(FACE .EQ. 2) NTI=N3/N2
      VALUE=NTI**2-(SIN(PHI)**2)
      IF(VALUE .LT. 0.0) GO TO 10
      SQ=SQRT(VALUE)
      COSP=COS(PHI)
      RPER=(COSP-SQ)/(COSP+SQ)
      RPAR=((NTI**2)*COSP-SQ)/((NTI**2)*COSP+SQ)
      XM1=1.0-0.5*(RPER**2+RPAR**2)
      RETURN
   10 CONTINUE
      FACE=3
      RETURN
      END
C
C
C    SUBROUTINE RADYUS CALCULATES THE RADIAL DIMENSION FROM THE CENTER
C     OF SYMMETRY TO THE RAY AT THE ANGLE SPECIFIED.
C     RADYUS ALSO FINDS D(R)/D(THETA) FOR SUBR ITRATE.
C
      SUBROUTINE RADYUS(A,E,ARG,RSINO,ANGLE,R,DR,FILTER,EPSLON)
      REAL EPSLON
      DOUBLE PRECISION ANGLE,R,DR,ARG,RSINO,FRACTN,A,E
      INTEGER J,FILTER
      FRACTN=DSIN(-2.0*EPSLON*ANGLE+RSINO)
      IF((A+ARG*FRACTN) .LE. 0.0) GO TO 10
      R=SQRT(2.0)*DABS(E)/DSQRT(A+ARG*FRACTN)
      DR=((EPSLON*R**3)/(2.0*E**2))*DCOS(-2.0*EPSLON*ANGLE+RSINO)*ARG
   10 CONTINUE
      FILTER=0
      IF((A+ARG*FRACTN) .LT. 0.0) FILTER=1
      RETURN
      END
C
C   SUBROUTINE RADUS CALCULATES THE RADIUS TO THE FRONT SURFACE AT
```

```
C     THE SPECIFIED ANGLE AND ALSO FINDS ITS DERIVATIVE WITH RESPECT
C     TO ANGLE.
C
      SUBROUTINE RADUS(PI,X1,Y1,I1,U,CSYMB,THETAP,RADD,RDG)
      DOUBLE PRECISION X1,Y1,I1,U,THETAP,RADD,RDG,PI,PI2,CSYMB
      PI2=PI/2.
      S=DCOTAN(I1+U)
      B=Y1-S*(X1+CSYMB)
      RADD=B/(DSIN(THETAP)-S*DCOS(THETAP))
      RDG=-B*(DCOS(THETAP)+S*DSIN(THETAP))/
     1((DSIN(THETAP)-S*DCOS(THETAP))**2)
      RETURN
      END
C
C     SUBROUTINE INTERCEPT1 PERFORMS ITERATION TO FIND R AND THETAP
C     OF THE INTERCEPT OF THE GRIN SKEW-RAY AND THE INSIDE CONICAL
C     SURFACE USING THE NEWTON-RAPHSON ITERATION PROCEDURE.
C
      SUBROUTINE INTER1(SIGN,THETA,R,THETAP,FILTER,A,E,ARG,RSINO,
     1EPSLON,C,TCRIT,NPOX,NPOY,NPOZ,PSIO,XO,YO,ZO,RO,CKKP,CLLP,
     1CMMP,ALPHA,CSYMB,FLUG)
      INTEGER LOOP,FILTER,FLUG
      REAL SIGN,EPSLON
      DOUBLE PRECISION THETA,R,THETAP,A,E,ARG,RSINO,RO,CSYMB,
     1ALPHA,RADG,RADH,FX,FXDOT,RDOTG,RDOTS,XNEW,DIF,C,TCRIT,PSIO,
     1FACTOR,EXIT,THET,PI,XO,YO,ZO,NPOX,NPOY,NPOZ,CKKP,CLLP,CMMP
C
      LOOP=0
      FILTER=0
      PI=3.141592653589793
   10 CONTINUE
      THET=THETA
      IF(THETA.GT.TCRIT)  THET=THETA-C
      IF(THETA.GT.TCRIT)  EPSLON=+1.0
      CALL RADYUS(A,E,ARG,RSINO,THET,RADG,RDOTG,FILTER,EPSLON)
      IF(FILTER .EQ. 1)RETURN
      IF(FILTER .EQ. 2)RETURN
      CALL SURF(THETA,RADH,RDOTS,SIGN,FILTER,NPOX,NPOY,NPOZ,PSIO,XO,
     1YO,ZO,CSYMB,RO,CKKP,CLLP,CMMP,ALPHA)
      IF(FILTER .EQ. 3)RETURN
      FX=RADG-RADH
      FXDOT=RDOTG-RDOTS
      FACTOR=1.7/(DABS(PI-PSIO))**.5
      IF(CSYMB.GT.0.0) FACTOR=1.3
      XNEW=THETA-(FX/FXDOT)/FACTOR
      DIF=DABS(THETA-XNEW)
      EXIT=0.0000001*DABS(PI-PSIO)
      IF(CSYMB.GT.0.0) EXIT=0.0000001
```

```
      IF(DIF .GT. EXIT) GO TO 20
      THETAP=THETA
      R=RADG
      RETURN
   20 CONTINUE
      IF(LOOP .GE. 90) GO TO 30
      LOOP=LOOP+1
      THETA=XNEW
      GO TO 10
   30 CONTINUE
      IF(LOOP .GE. 90) FILTER=3
      WRITE(6,6410)
 6410 FORMAT(/1X,'SOLUTION NOT FOUND')
      RETURN
      END
C
C     SUBROUTINE INTERCEPT2 PERFORMS ITERATION TO FIND THE INTERSECTION
C     OF THE GRIN RAY AND THE INSIDE SURFACE USING THE ANGLE IN
C     TERMS OF RADIUS PROCEDURE.
C
      SUBROUTINE INTER2(RP,THETAP,RPP,THATP,A,E,ARG,RSINO,EPSLON,
     1SIGN,NPOX,NPOY,NPOZ,PSIO,XO,YO,ZO,CSYMB,RO,CKKP,CLLP,CMMP,
     1ALPHA,THETA,FILTER,FLUG)
      INTEGER TRIP,FILTER,FLUG
      REAL SIGN,EPSLON
      DOUBLE PRECISION RP,THETAP,RPP,A,E,ARG,RSINO,CSYMB,
     1PSIO,RO,ALPHA,RSIN,CHECK1,CHECK2,THETA,THATP,RDOT,ARGH,
     1XO,YO,ZO,NPOX,NPOY,NPOZ,CKKP,CLLP,CMMP
      FILTER=0
      TRIP=0
   10 TRIP=TRIP+1
      ARGH=(2*E**2/RP**2-A)/ARG
      IF(DABS(ARGH).GT.1.0) GO TO 50
      RSIN=DARSIN(ARGH)
      THETPP=-EPSLON/2*(RSIN-RSINO)
      CHECK1=THETPP-THETAP
      CALL SURF(THETPP,RPP,RDOT,SIGN,FILTER,NPOX,NPOY,NPOZ,PSIO,XO,
     1YO,ZO,CSYMB,RO,CKKP,CLLP,CMMP,ALPHA)
      CHECK2=DABS(RP-RPP)
      RP=RPP
      THETAP=THETPP
      IF(TRIP.GT.100) GO TO 20
      IF(CHECK2.LT.0.000001) GO TO 20
      GO TO 10
   20 THATP=THETAP
   50 RETURN
      END
C
```

```
C      SUBROUTINE SURF CALCULATES BOTH THE RADIUS TO THE
C     LOCUS OF THE INTERCEPT OF THE RAY PLANE AND THE IN-
C     SIDE SURFACE AND THE DERIVATIVE OF THE RADIUS WRT THETA
C     GIVEN THE ANGLE THETA. SURF IS DESIGNED PRIMARILY
C     FOR USE WITH SUBROUTINES INTERCEPT1 AND INTERCEPT2.
C
      SUBROUTINE SURF(THETA,R,RDOT,SIGN,FILTER,NPOX,NPOY,NPOZ,PSIO,XO,
     1YO,ZO,CSYMB,RO,CKKP,CLLP,CMMP,ALPHA)
      INTEGER FILTER
      REAL XO,YO,ZO,NPOX,NPOY,NPOZ,CKKP,CLLP,CMMP,SIGN
      DOUBLE PRECISION THETA,R,RDOT,PSIO,CSYMB,
     1RO,ALPHA,ALFA,BETA,A,B,C,A2,B2,C2,SQUARE,
     1DAL,DBE,DA,DB,DC,DA2,DB2,A3,B3,C3
      ALFA=DCOS(THETA)-DSIN(THETA)*DCOTAN(PSIO)
      BETA=DSIN(THETA)/DSIN(PSIO)
      A=ALFA*(XO+CSYMB)/RO+BETA*CKKP
      B=ALFA*YO/RO+BETA*CLLP
      C=ALFA*ZO/RO+BETA*CMMP
      A2=B**2+C**2-(A**2)*(DTAN(ALPHA)**2)
      B2=B*NPOY+C*NPOZ+A*NPOX+2.0*A*CSYMB*(DTAN(ALPHA)**2)
      C2=CSYMB*NPOX+(CSYMB*DTAN(ALPHA))**2+NPOX*XO+NPOY*YO+NPOZ*ZO
      SQUARE=0.25*(B2/A2)**2+C2/A2
      IF(SQUARE .LT. 0.0) FILTER=3
      IF(FILTER .EQ. 3) RETURN
      R=-B2/(2.0*A2)+SIGN*DSQRT(SQUARE)
C     CALCULATE DR/DTHETA
      DAL=-DSIN(THETA)-DCOS(THETA)*DCOTAN(PSIO)
      DBE=DCOS(THETA)/DSIN(PSIO)
      DA=DAL*(XO+CSYMB)/RO+DBE*CKKP
      DB=DAL*YO/RO+DBE*CLLP
      DC=DAL*ZO/RO+DBE*CMMP
      DA2=2.0*B*DB+2.0*C*DC-2.0*A*DA*(DTAN(ALPHA)**2)
      DB2=DB*NPOY+DC*NPOZ+DA*NPOX+2.0*DA*CSYMB*(DTAN(ALPHA)**2)
      A3=-0.5*(A2*DB2-B2*DA2)/(A2**2)
      B3=SIGN/(2.0*DSQRT(0.25*(B2/A2)**2+C2/A2))
      C3=0.25*((A2**2)*2.0*B2*DB2-(B2**2)*2.0*A2*DA2)/(A2**4)-
     1C2*DA2/(A2**2)
      RDOT=A3+B3*C3
      RETURN
      END
C
C
C     SUBROUTINE DIRECT CALCULATES THE VECTOR DIRECTION COSINES OF
C      A SKEW RAY AT THE POINT OF INTERCEPT WITH THE INSIDE SURFACE.
C
      SUBROUTINE DIRECT(CKP,CLP,CMP,DIR,RRX,RRY,RRZ,NPFX,NPFY,NPFZ,PSIR)
      DOUBLE PRECISION CKP,CLP,CMP,DIR,RRX,RRY,RRZ,NPFX,NPFY,NPFZ,PSIR,
     1EPS,APP,BPP,CPP,AA,BB,CC,AP,BP,CP
```

```
      EPS=1.0E-06
      SIGN=+1.0
      IF((DABS(NPFX)) .GT. EPS) GO TO 50
      IF((DABS(NPFY)) .GT. EPS) GO TO 30
      IF((DABS(RRY)) .LT. EPS) GO TO 25
         CMP=0.0
      APP=1.0+(RRX/RRY)**2
      BPP=DCOS(PSIR)*RRX/RRY**2
      CPP=(DCOS(PSIR)/RRY)**2-1.0
         CKP=BPP/APP+SIGN*DSQRT(BPP**2-APP*CPP)/APP
         CLP=(DCOS(PSIR)-RRX*CKP)/RRY
         RETURN
   25 CONTINUE
         CMP=0.0
         CKP=DCOS(PSIR)/RRX
         CLP=SIGN*DSQRT(1.0-CKP**2)
         RETURN
   30 CONTINUE
      IF((DABS(NPFZ)) .GT. EPS) GO TO 40
      IF((DABS(RRZ)) .LT. EPS) GO TO 35
         CLP=0.0
      APP=1.0+(RRX/RRZ)**2
      BPP=DCOS(PSIR)*RRX/RRZ**2
      CPP=(DCOS(PSIR)/RRZ)**2-1.0
         CKP=BPP/APP+DSQRT(BPP**2-APP*CPP)/APP
         CMP=(DCOS(PSIR)-RRX*CKP)/RRZ
         RETURN
   35 CONTINUE
         CLP=0.0
         CKP=DCOS(PSIR)/RRX
         CMP=SIGN*DSQRT(1.0-CKP**2)
         RETURN
   40 CONTINUE
      IF((DABS(RRX)) .GT. EPS) GO TO 45
         CLP=DCOS(PSIR)/(RRY-NPFY/NPFZ*RRZ)
         CMP=-NPFY/NPFZ*CLP
         CKP=SIGN*DSQRT(1.0-CLP**2-CMP**2)
         RETURN
   45 CONTINUE
      IF((DABS(RRY-NPFY/NPFZ*RRZ)) .GT. EPS) GO TO 47
         CKP=DCOS(PSIR)/RRX
         CLP=SIGN*DSQRT((1.0-CKP**2)/(1.0+(NPFY/NPFZ)**2))
         CMP=-NPFY/NPFZ*CLP
         RETURN
   47 CONTINUE
      SIGN=-1.0
      IF(RRY .LT. 0.0 .OR. RRX .LT. 0.0) SIGN=+1.0
      IF(RRY .LT. 0.0 .AND. RRX .LT. 0.0) SIGN=-1.0
```

```
      APP=((RRY-NPFY/NPFZ*RRZ)/RRX)**2+1.0+(NPFY/NPFZ)**2                GIS12970
      BPP=2.0*DCOS(PSIR)/(RRX**2)*(RRY-NPFY/NPFZ*RRZ)                   GIS12980
      CPP=(DCOS(PSIR)/RRX)**2-1.0                                       GIS12990
        CLP=BPP/(2.0*APP)+SIGN*DSQRT(BPP**2-4.0*APP*CPP)/(2.0*APP)     GIS13000
        CMP=-NPFY/NPFZ*CLP                                              GIS13010
        CKP=(DCOS(PSIR)-(RRY-NPFY/NPFZ*RRZ)*CLP)/RRX                    GIS13020
        RETURN                                                          GIS13030
   50 CONTINUE                                                          GIS13040
      SIGN=-1.0                                                         GIS13050
      IF(RRY .LT. 0.0)SIGN=+1.0                                         GIS13060
      IF(RRY .LT. 0.0 .AND. NPFZ .LT. 0.0) SIGN=-1.0                    GIS13070
      AA=NPFX*RRY-NPFY*RRX                                              GIS13080
      BB=NPFX*RRZ-NPFZ*RRX                                              GIS13090
      CC=NPFX*DCOS(PSIR)                                                GIS13100
      IF((DABS(BB)) .GT. EPS)GO TO 60                                   GIS13110
        CLP=CC/AA                                                       GIS13120
      APP=(NPFZ/NPFX)**2+1.0                                            GIS13130
      BPP=NPFY*NPFZ/NPFX**2*CLP                                         GIS13140
      CPP=((NPFY/NPFX)**2+1.0)*CLP**2-1.0                               GIS13150
        CMP=-BPP/APP+SIGN*DSQRT(BPP**2-APP*CPP)/APP                     GIS13160
        CKP=(-NPFY*CLP-NPFZ*CMP)/NPFX                                   GIS13170
        RETURN                                                          GIS13180
   60 CONTINUE                                                          GIS13190
      IF(BB .GT. 0.0) SIGN=-1.0                                         GIS13200
      AP=(NPFY/NPFX)**2+1.0                                             GIS13210
      BP=(NPFZ/NPFX)**2+1.0                                             GIS13220
      CP=2.0*NPFY*NPFZ/NPFX**2                                          GIS13230
      APP=AP+BP*(AA/BB)**2-CP*AA/BB                                     GIS13240
      BPP=2.0*AA*CC*BP/BB**2-CP*CC/BB                                   GIS13250
      CPP=BP*(CC/BB)**2-1.0                                             GIS13260
        CLP=BPP/(2.0*APP)+SIGN*DSQRT(BPP**2-4.0*APP*CPP)/(2.0*APP)     GIS13270
        CMP=CC/BB-AA/BB*CLP                                             GIS13280
        CKP=(-NPFY*CLP-NPFZ*CMP)/NPFX                                   GIS13290
        RETURN                                                          GIS13300
      END                                                               GIS13310
```

```
C       ********************************************************************
C       *                                                                  *
C       *                          C H E C K                               *
C       *                                                                  *
C       ********************************************************************
C       *                                                                  *
C       *          THIS PROGRAM IS DESIGNED TO CHECK THE ACCURACY OF       *
C       *      THE GISL FRONT SURFACE.   IT DOES THIS BY COMPLETING A      *
C       *      RAY TRACE FROM THE MIDPOINT OF EACH SET OF POINTS THAT      *
C       *      DEFINE THE FRONT SURFACE, THROUGH THE LENS AND TOWARDS      *
C       *      THE FOCAL POINT.   DEVIATIONS IN THE Y-COORDINATE AT        *
C       *      THE FOCAL POINT ARE RECORDED.                               *
C       ********************************************************************
C
C       MAJOR VARIABLE DEFINITIONS:
C
C          XO,YO     : COORDINATES OF MIDPOINT AT FRONT SURFACE
C          THETAO    : INITIAL ANGLE
C          RO        : INITIAL RADIUS
C          PSIO      : INITIAL ANGLE BETWEEN RADIUS AND RAY DIRECTION
C          C         : RAY MATCHING CONSTANT
C          THETAC    : ANGLE OF MINIMUM RADIUS
C          XI,YI     : FIRST GUESS COORDINATES AT INNER SURFACE
C          XF,YF     : SOLUTION POINT COORDINATES
C          THETAP    : ANGLE AT INNER SURFACE
C          RAD       : RADIUS AT INNER SURFACE
C          PSIP      : ANGLE BETWEEN RADIUS AND RAY DIRECTION AT
C                        INNER SURFACE
C          YS        : Y-COORDINATE ON IMAGE PLANE
C
C
C       SPECIFICATION STATEMENTS
C
        INTEGER K,I,L,TRIP,EXIT2
C
        DOUBLE PRECISION DIFF,THETAN,THETAP,THETAH,
      1F,ALPHA,RL,N1,N2,N3,N20,OSYMB,A,B,PI,PI2,BF,RZERO,
      1X1(1010),Y1(1010),X2(1010),Y2(1010),X20,Y20,RZ,XO,YO,XI,YI,
      1I1,I1P,I2,I2P,THETAO,RO,ARG,RSINO,C,TCRIT,PSIO,E,EPLSON,
      1S,S1,S2,S3,XF,YF,PSIP,ZETA,UP,YS(1010),SIN,EON,EONR,RP,RPP,BE,
      1SL,RSIN,THETA,CHECK1,CHECK2
C
C  READ INPUT FROM PROGRAM GISL
C
        READ(5,100)A,B,OSYMB,K,ALPHA,N3,F,RL
        DO 5 J=1,K
        READ(5,200)X1(J),Y1(J),X2(J),Y2(J)
    5   CONTINUE
```

```
C
C   SET CONSTANTS
C
      PI=3.141592653589793
      PI2=PI/2
      N1=1.0
      N3=1.0
      EF=F/RL
      RZERO=((X2(1)+DSYMB)**2+Y2(1)**2)**.5
      IF(DSYMB.LT.-0.6) RZERO=DSQRT(R**2+DSYMB**2)
C
C   START RAY TRACE
C
      DO 10 J=2,K
      L=J-1
      TRIP=0
C
C   CALCULATE MIDPOINT
C
      S=(Y1(J)-Y1(L))/(X1(J)-X1(L))
      XO=(X1(J)+X1(L))/2.
      YO=(Y1(J)+Y1(L))/2.
C
C   CALCULATE LINE CONSTANTS
C
      I1=PI2-DATAN(S)
      THETAO=DATAN(YO/(XO+DSYMB))
      IF(THETAO.LT.0.0) THETAO=PI+THETAO
      RO=DSQRT(YO**2+(XO+DSYMB)**2)
      CALL INDEX(RO,N20,A,B,RZERO)
      I1P=DARSIN((N1/N20)*DSIN(I1))
      PSIO=PI-THETAO-I1+I1P
      EPSLON=1.0
      IF(PSIO.GT.PI2.OR.PSIO.LT.0.0) EPSLON=-1.0
      E=RO*N20*DSIN(PSIO)
      ARG=DSQRT(A**2+4*B*(E**2)/(RZERO**2))
      RSINO=DARSIN((2*(E**2)/(RO**2)-A)/ARG)
      C=PI2-RSINO
      TCRIT=THETAO-(EPSLON/2.)*(PI2-RSINO)
      IF(PSIO.LT.PI2) C=0.0
C
C   FIND FIRST GUESS COORDINATES, RADIUS AND ANGLE
C
      S1=-DTAN(I1-I1P)
      S2=DTAN(ALPHA)
      XI=(YO-S1*XO)/(S2-S1)
      YI=S2*XI
      THETAP=DATAN(YI/(XI+DSYMB))
```

```
      IF(THETAP.LT.0.0) THETAP=THETAP+PI
      RP=CSQRT(YI**2+(XI+DSYMB)**2)
      EE=-DSYME
      SL=1.0
C
C  GO TO THETA IN TERMS OF RADIUS ITERATION IF ANGLE LARGE
C
      IF(THETAP.GT.2.0) GO TO 25
C
C  NEWTON-RAPHSON ITERATION
C
   20 CONTINUE
         TRIP=TRIP+1
         IF(THETAP.GE.TCRIT) THETP=THETAP-C
         IF(THETAP.LT.TCRIT) THETP=THETAP
         IF(THETAP.GE.TCRIT) EPSLON=+1.0
         THETAH=THETP-THETAO
         CALL RADYUS(ARG,RSINO,THETAH,RAD,RDG,FILTER,A,EPSLON,E)
         CALL RADUS(CSYMB,S2,THETAP,RADD,RDGG)
         RDF=RAD-RADD
         RDT=RDG-RDGG
         FACTOR=1.7/(DABS(PSIO))
         IF(FACTOR.GT.10.0) FACTOR=10.0
         IF(CSYMB.GT.0.0) FACTOR=1.3
         THETAN=THETAP-RDF/(RDT*FACTOR)
         DIFF=DABS(THETAN-THETAP)
         EXIT=0.00001*DABS(PSIO)
         IF(CSYMB.GT.0.0) EXIT=0.000001
         EXIT2=200
         IF(DIFF.LT.EXIT) GO TO 30
         IF(TRIP.GT.EXIT2) GO TO 30
         THETAP=THETAN
         GO TO 20
C
C  THETA IN TERMS OF RADIUS ITERATION
C
   25    TRIP=TRIP+1
         RSIN=DARSIN((2*E**2/RP**2-A)/ARG)
         THETA=THETAO-EPSLON/2*(RSIN-RSINO)
         CHECK1=THETA-THETAP
         RPP=EE/(DSIN(THETA)-SL*DCOS(THETA))
         CHECK2=DABS(RP-RPP)
         THETAP=THETA
         RP=RPP
         IF(TRIP.GT.100) GO TO 26
         IF(CHECK2.LT.0.0000001) GO TO 26
         GO TO 25
   26    RAD=RPP
```

```
C
C     CALCULATE FINAL POINT AT INNER SURFACE COORDINATES AND ANGLES
C
   30 XF=RAD*DCOS(THETAP)-OSYMB
      YF=RAD*DSIN(THETAP)
      CALL INDEX(RAD,N2,A,B,RZERO)
      EONR=E/(N2*RAD)
      IF(EONR.GT.1.0) WRITE(6,630)
      IF(EONR.GT.1.0) EONR=1.0
      PSIP=DARSIN(EONR)
      IF(THETAP.LT.TCRIT.AND.PSIO.GT.0.0) PSIP=PI-PSIP
      ZETA=PI-THETAP-PSIP
      I2=PI2-ZETA-ALPHA
      EON=(N2/N3)*DSIN(I2)
      IF(EON.GT.1.0) WRITE(6,640)
      IF(EON.GT.1.0) EON=1.0
      I2P=DARSIN(EON)
      UP=PI2-ALPHA-I2P
C
C     CALCULATE Y-COORDINATE AT FOCAL POINT
C
      S3=-DTAN(UP)
      YS(J)=S3*(F-XF)+YF
C
C     WRITE OUTPUT
C
      WRITE(6,500) J,PSIO,YS(J),TRIP
   10 CONTINUE
      STOP
C
C     FORMAT STATEMENTS
C
  100 FORMAT(1X,3F11.7,I5,F10.7,3F5.3)
  200 FORMAT(1X,4F10.7)
  500 FORMAT(1X,I6,2F16.8,I6)
  630 FORMAT(1X,'WHOOPS PSIP ARSIN')
  640 FORMAT(1X,'WHOOPS AGAIN I2P ARSIN')
C
      END
C
C        SUBROUTINE INDEX CALCULATES THE INDEX OF REFRACTION AS A FUNCTION
C         OF RADIUS FROM THE CENTER OF SYMMETRY. THE VALUES OF THE INDEX
C         CONSTANTS A AND B ARE USER INPUTS.
C
      SUBROUTINE INDEX(R,N,A,B,RZERO)
      DOUBLE PRECISION N,A,B,R,RZERO
      N=DSQRT(A+B*(R/RZERO)**2)
      RETURN
```

```
      END
C
C     SUBROUTINE RADYUS CALCULATES THE RADIUS FOR THE GRIN EQUATION AT
C      THE SPECIFIED ANGLE AS WELL AS ITS DERIVATIVE WITH RESPECT TO
C      ANGLE.
C
      SUBROUTINE RADYUS(ARG,RSINO,ANGLE,R,DR,FILTER,A,EPSLON,E)
      INTEGER FILTER
      DOUBLE PRECISION ANGLE,R,DR,ARG,RSINO,A,EPSLON,E,FRACTN
      FRACTN=DSIN(-2.0*EPSLON*ANGLE+RSINO)
      IF((A+ARG*FRACTN) .LE. 0.0) GO TO 10
      R=SQRT(2.0)*DABS(E)/DSQRT(A+ARG*FRACTN)
      IF(FRACTN .LE. (-0.989)) GO TO 10
      DR=((EPSLON*R**3)/(2.0*E**2))*DCOS(-2.0*EPSLON*ANGLE+RSINO)
   10 CONTINUE
      FILTER=0
      IF(FRACTN .LE. (-0.989))FILTER=2
      IF((A+ARG*FRACTN) .LT. 0.0) FILTER=1
      RETURN
      END
C
C    SUBROUTINE RADUS CALCULATES THE RADIUS TO THE FRONT SURFACE AT
C     THE SPECIFIED ANGLE AND ALSO THE DERIVATIVE WITH RESPECT TO
C     ANGLE.
C
      SUBROUTINE RADUS(DSYMB,S,THETAP,RADD,RDGG)
      DOUBLE PRECISION THETAP,DSYMB,S
      RADD=-DSYMB*S/(DSIN(THETAP)-S*DCOS(THETAP))
      RDGG=S*DSYMB*(DCOS(THETAP)+S*DSIN(THETAP))/
     1((DSIN(THETAP)-S*DCOS(THETAP))**2)
      RETURN
      END
```

```
C ***************************************************************
C *                                                             *
C *              D E T P L C T   F O R T R A N                  *
C *                                                             *
C ***************************************************************
C *                                                             *
C *       THIS PROGRAM IS DESIGNED TO COMPLETE THE RAY TRACE OF *
C *    A MIRROR AND DETECTOR SYSTEM.   RAYS THAT LEAVE THE LENS *
C *    ARE TRACED TO THE MIRROR, THEN REFLECTED OFF AND TRACED  *
C *    TO THE DETECTOR.   THIS PROCESS IS REPEATED FOR EACH     *
C *    INCREMENT OF MIRROR ANGLE.   THE NUMBER OF RAYS STRIKING *
C *    THE UPPER AND LOWER HALF OF THE DETECTOR ARE RECORDED AT *
C *    EACH ANGLE AND THEN THIS INFORMATION IS OUTPUT SO THAT   *
C *    APPROPRIATE PLOTS MAY BE MADE.                           *
C *                                                             *
C ***************************************************************
C
C   VARIABLE DEFINITIONS
C
C      BF      : DISTANCE FROM FOCAL POINT TO CONE APEX
C      ALPHA   : CONE ANGLE OF BACK LENS SURFACE
C      ALFAP   : ANGLE OF RADIATION INCIDENT TO LENS
C      DETP    : DETECTOR POSITION
C      DETS    : DETECTOR RADIAL SIZE
C      MIRP    : MIRROR PIVOT POSITION
C      XC,YC,ZC     : COORDINATES OF RAY EXITING LENS
C      XM,YM,ZM     : COORDINATES OF RAY AT MIRROR SURFACE
C      YD,ZD        : COORDINATES OF RAY AT DETECTOR (XD=DETP)
C      CK,CL,CM     : DIRECTION COSINES OF RAY FROM LENS TO MIRROR
C      CKK,CLL,CMM: DIRECTION COSINES OF RAY FROM MIRROR TO DETECTOR
C      COUNT1       : RAY COUNT ON UPPER DETECTOR
C      COUNT2       : RAY COUNT ON LOWER DETECTOR
C
C
C   SPECIFICATION STATEMENTS
C
      INTEGER J,K,L,OUT1,OUT2(500),OUT3(500),COUNT1(500),COUNT2(500),
     1TERM1,TERM2
      REAL BF,ALPHA,PI,PI2,DETP,DETS,MIRP,THETM(200),XU(500),YO(500),
     1ZO(500),CK(500),CL(500),CM(500),D1,D2,XM(500),YM(500),ZM(500),
     1CKK,CLL,CMM,YD(500),ZD(500),ALFAP,THETD(500)
C
C   READ IN LENS CONSTANTS
C
      READ (5,100) BF,ALPHA,ALFAP
C
C
C
```

```
C     SET CONSTANTS
C
        PI=3.14159
        PI2=PI/2.
        DETP=0.2
        MIRP=1.4
        THETC=66.0*PI/180
        DETS=DETP*TAN(ALPHA)
        THETM(1)=THETC-0.20404040
        J=0
        L=1
        OUT1=0
        TERM1=0
        TERM2=0
C
C     READ IN RAY DATA REJECTING RAYS BLOCKED BY DETECTOR
C
   10   J=J+1
        READ(5,200) XO(J),YO(J),ZO(J),CK(J),CL(J),CM(J)
        IF(SQRT(YO(J)**2+ZO(J)**2).LT.DETS) GO TO 11
        IF(XO(J).EQ.1.0.AND.YO(J).EQ.1.0.AND.ZO(J).EQ.1.0) TERM1=J
        IF(XO(J).EQ.1.0.AND.YO(J).EQ.1.0.AND.ZO(J).EQ.1.0) GO TO 12
        GO TO 10
   11   J=J-1
        OUT1=OUT1+1
        GO TO 10
   12   CONTINUE
C
C     INCREMENT MIRROR ANGLE AND CHECK FOR COMPLETED SWEEP
C
   20   L=L+1
        K=L-1
        THETM(L)=THETM(K)+0.00404040
        THETD(L)=THETM(L)*180/PI
        IF(THETM(L).GT.(THETC+0.20)) TERM2=L-1
C       WRITE(6,2000) THETM(L),OUT1,TERM1,TERM2
        IF(THETM(L).GT.(THETC+0.20)) GO TO 40
        OUT2(L)=0
        OUT3(L)=0
        COUNT1(L)=0
        COUNT2(L)=0
C
C
C     RAY TRACE FROM BACK LENS SURFACE TO MIRROR AND THEN TO DETECTOR
C
C
        J=0
   30   J=J+1
```

```
      IF(J.EQ.TERM1) GO TO 20
C
C    FIND MIRROR INTERSECTION POINT  DISCARD RAYS OUTSIDE MIRROR
C
      D1=(TAN(THETM(L))*(XO(J)-MIRP)-YO(J))/(CL(J)-CK(J)*TAN(THETM(L)))
      XM(J)=D1*CK(J)+XO(J)
      YM(J)=D1*CL(J)+YO(J)
      ZM(J)=D1*CM(J)+ZO(J)
C     WRITE(6,1000) XM(J),YM(J),ZM(J)
      IF(SQRT(ZM(J)**2+(COS(THETM(L))*YM(J))**2).GT.1.0)
     1OUT2(L)=OUT2(L)+1
      IF(SQRT(ZM(J)**2+(COS(THETM(L))*YM(J))**2).GT.1.0) GO TO 30
C
C    FIND DIRECTION OF RAY REFLECTED FROM MIRROR SURFACE
C
      CKK=+COS(2*THETM(L))*CK(J)+SIN(2*THETM(L))*CL(J)
      CLL=-COS(2*THETM(L))*CL(J)+SIN(2*THETM(L))*CK(J)
      CMM=CM(J)
C     WRITE(6,1010) CK(J),CL(J),CM(J),CKK,CLL,CMM
C
C    FIND INTERSECTION WITH DETECTOR  DISCARD RAYS MISSING DETECTOR
C
      D2=(DETP-XM(J))/CKK
      YD(J)=D2*CLL+YM(J)
      ZD(J)=D2*CMM+ZM(J)
      IF(SQRT(YD(J)**2+ZD(J)**2).GT.DETS) OUT3(L)=OUT3(L)+1
      IF(SQRT(YD(J)**2+ZD(J)**2).GT.DETS) GO TO 30
C
C    COUNT RAYS HITTING UPPER AND LOWER PORTIONS OF DETECTOR
C
      IF(YD(J).GT.0.0) COUNT1(L)=COUNT1(L)+1
      IF(YD(J).LT.0.0) COUNT2(L)=COUNT2(L)+1
      GO TO 30
C
C    WRITE OUTPUT
C
   40 CONTINUE
C     WRITE(6,2005) ALFAP
      DO 50 L=2,TERM2
      WRITE(6,2010) THETD(L),COUNT1(L),COUNT2(L)
   50 CONTINUE
      STOP
C
C    FORMAT STATEMENTS
C
  100 FORMAT(1X,3F10.7)
  200 FORMAT(1X,6F10.7)
C1000 FORMAT(1X,3F10.6)
```

```
C101C FORMAT(1X,6F10.6)
C200C FURMAT(1X,F10.7,3I10)
C2005 FURMAT(1X,F10.6)
 201C FURMAT(1X,F10.6,2I5)
C
      ENC
```

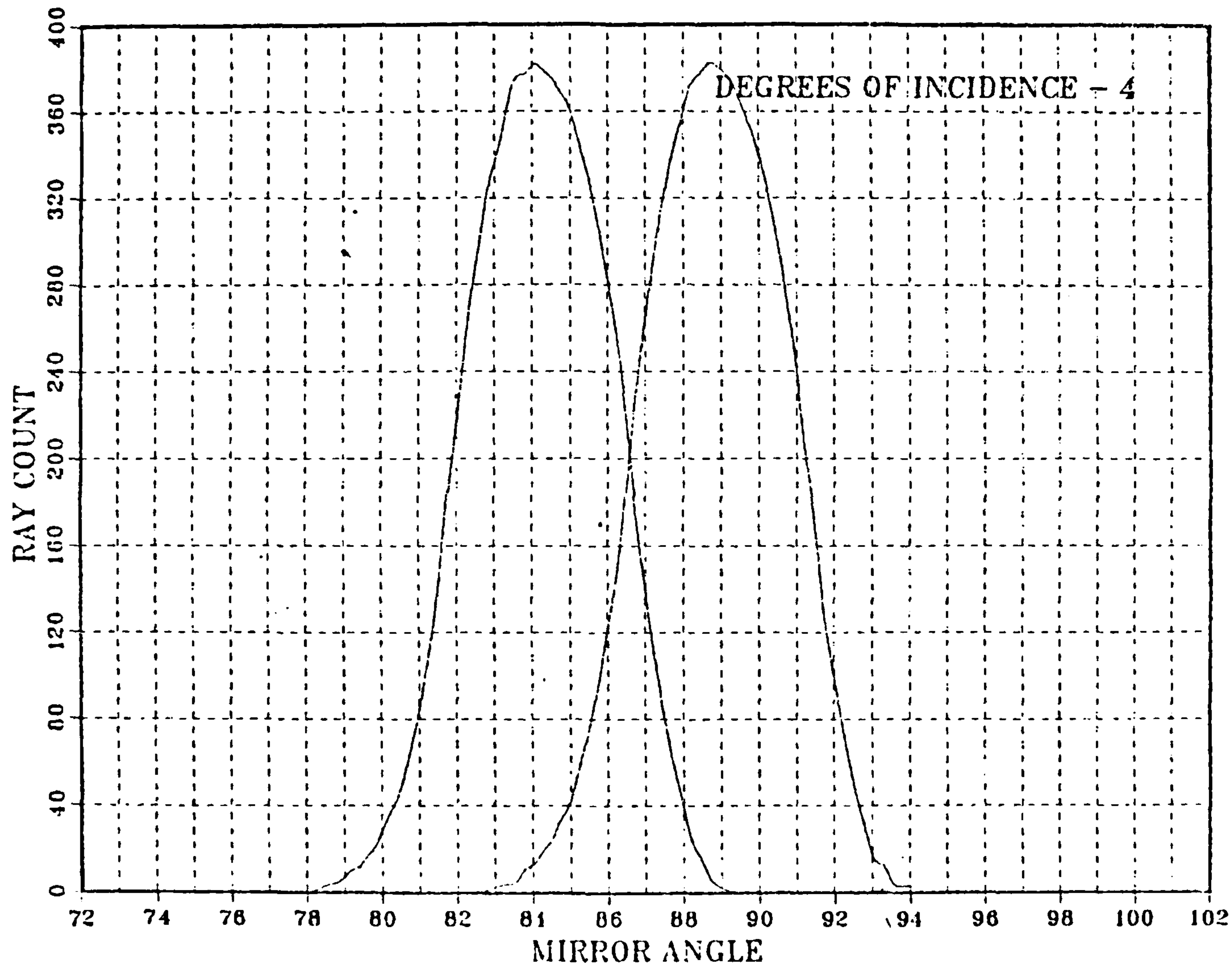

Figure D-1. Detector Signal at 4 Degrees Incidence.

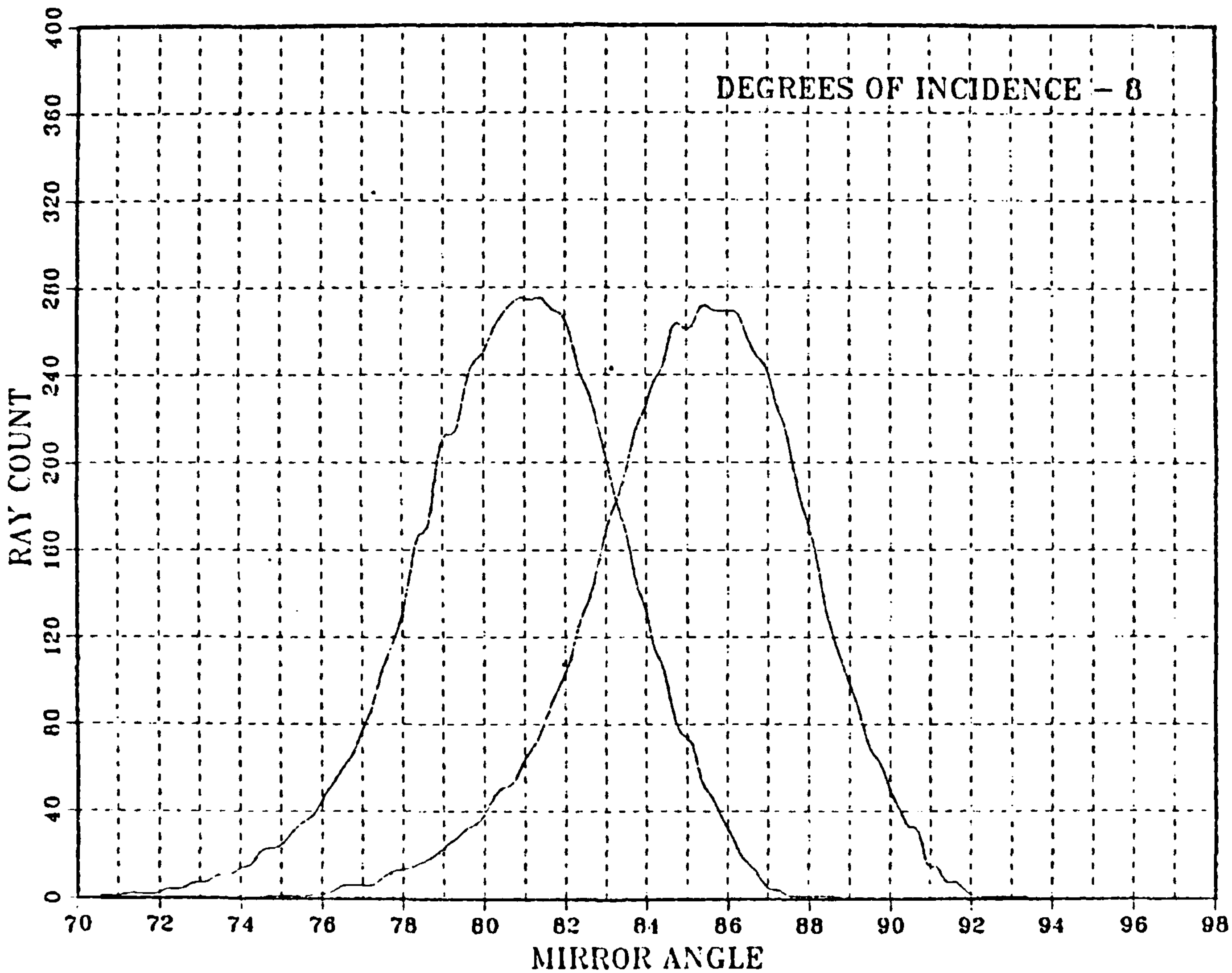

Figure D-2. Detector Signal at 8 Degrees Incidence.

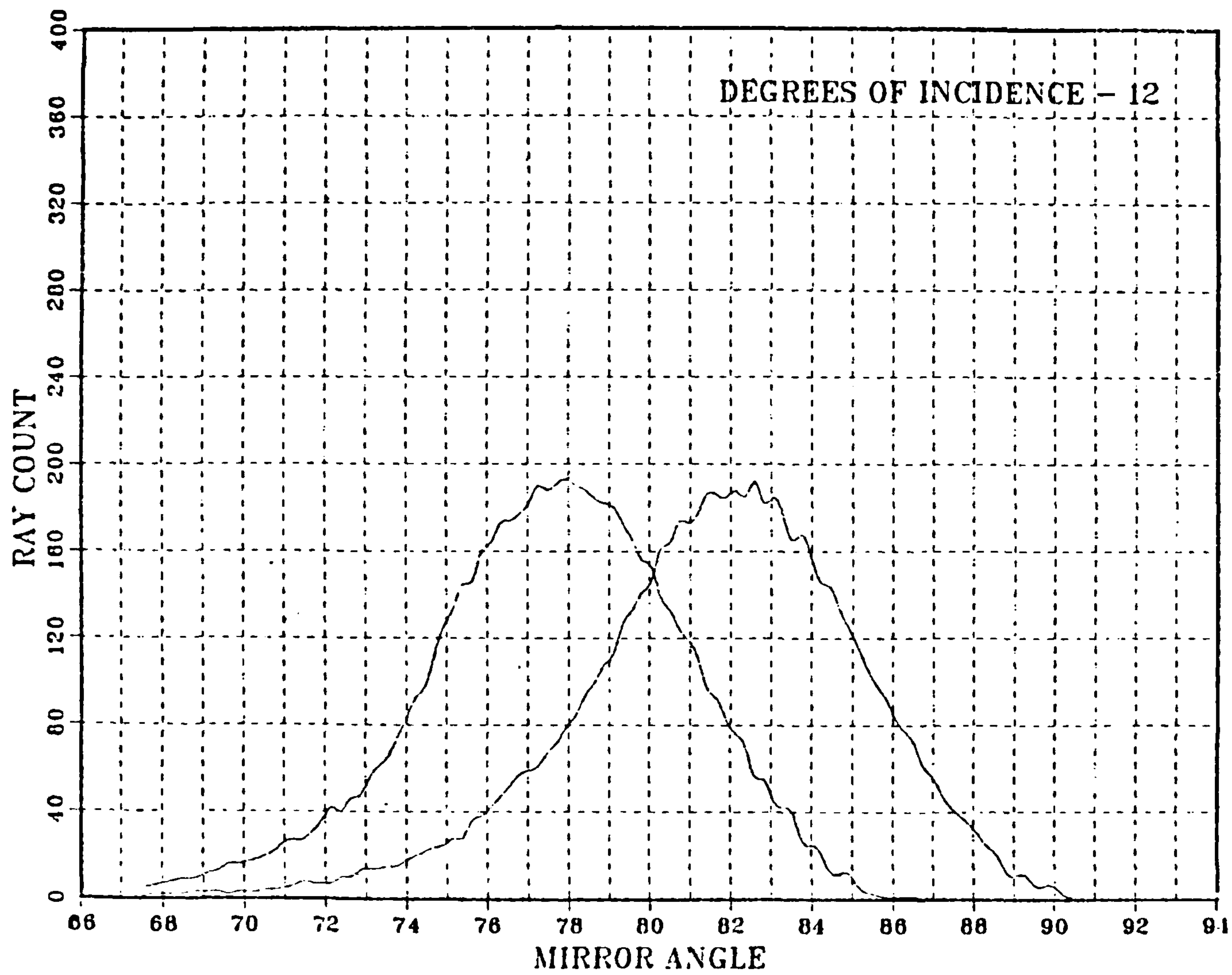

Figure D-3. Detector Signal at 12 Degrees Incidence.

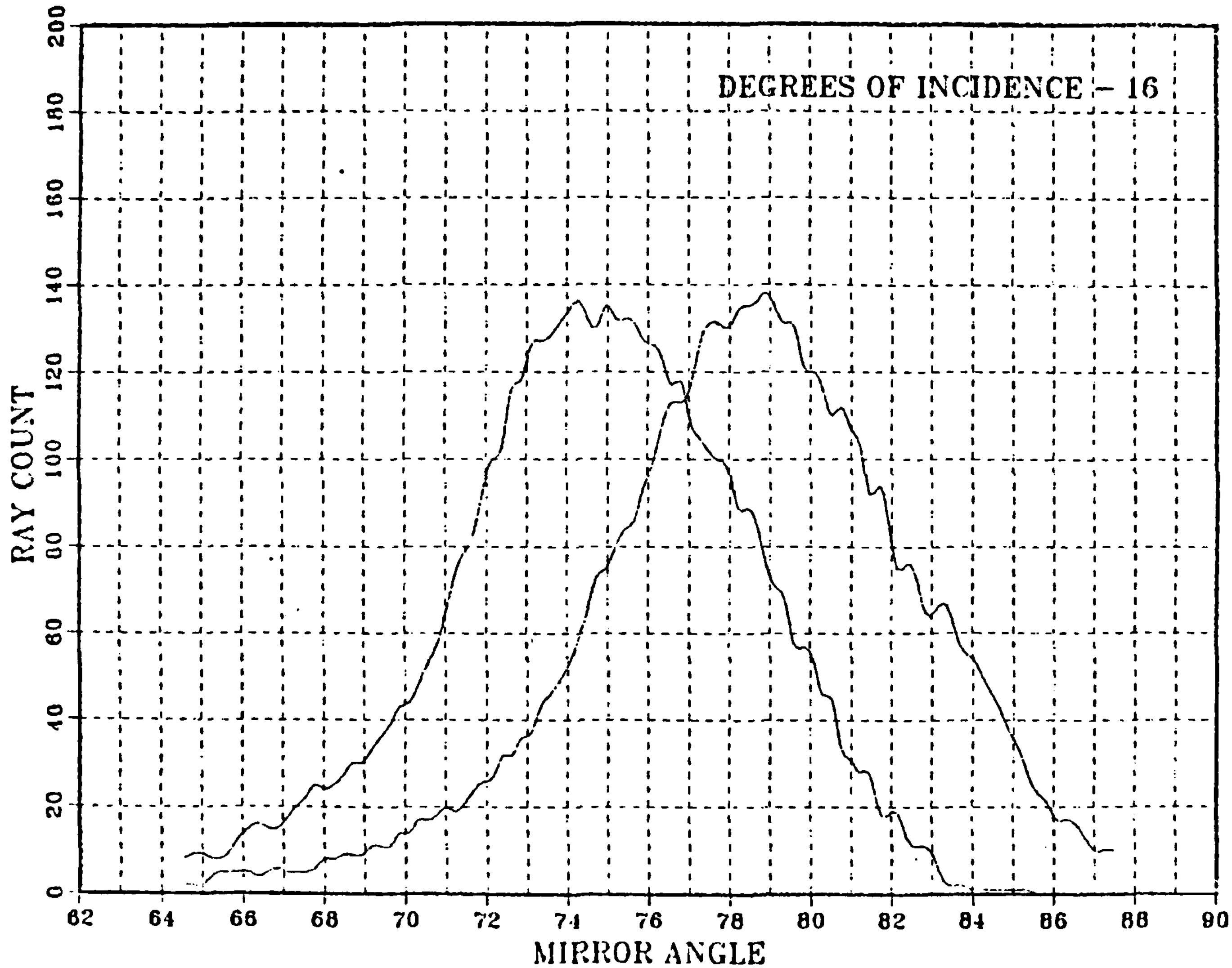

Figure D-4. Detector Signal at 16 Degrees Incidence.

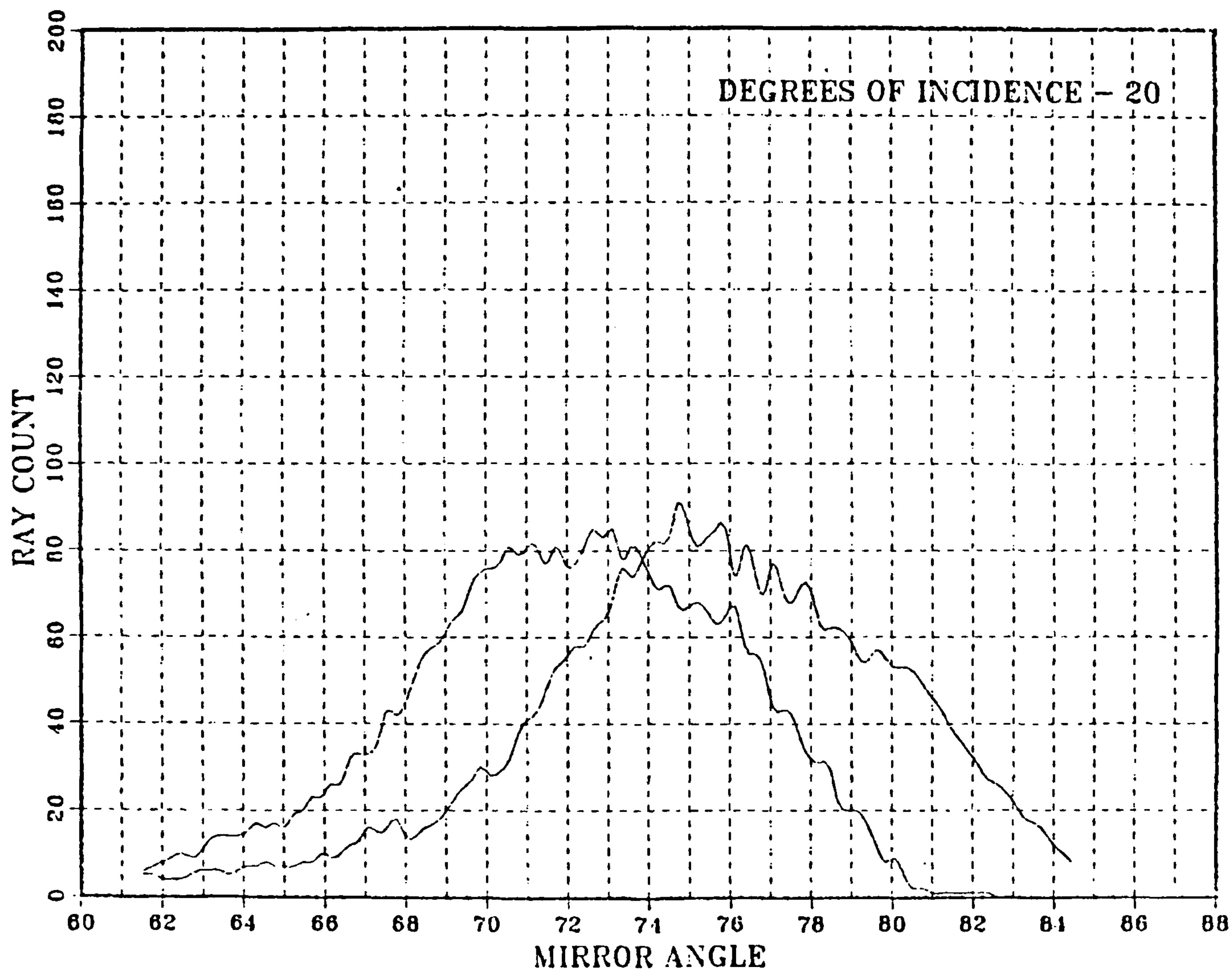

Figure D-5. Detector Signal at 20 Degrees Incidence.

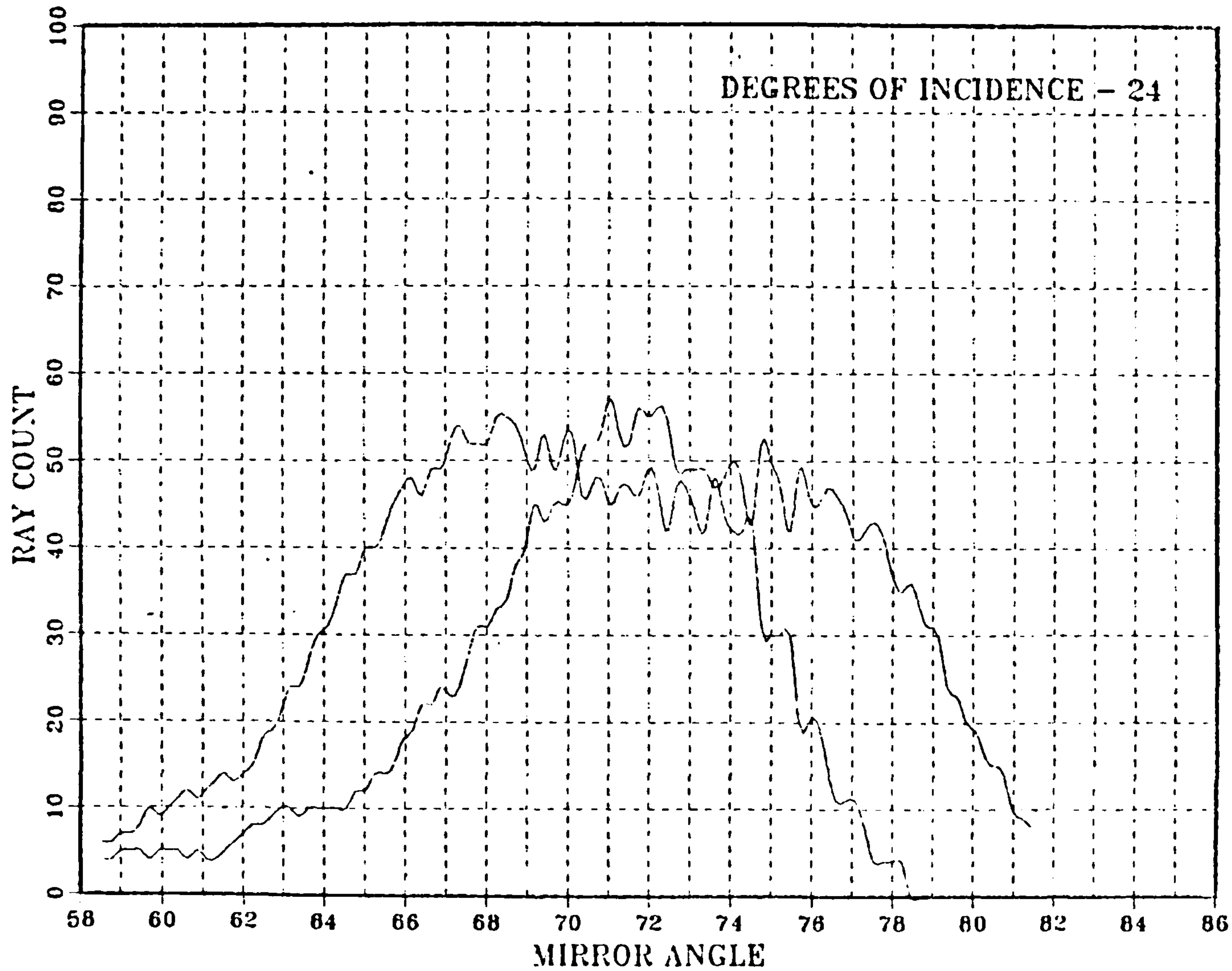

Figure D-6. Detector Signal at 24 Degrees Incidence.

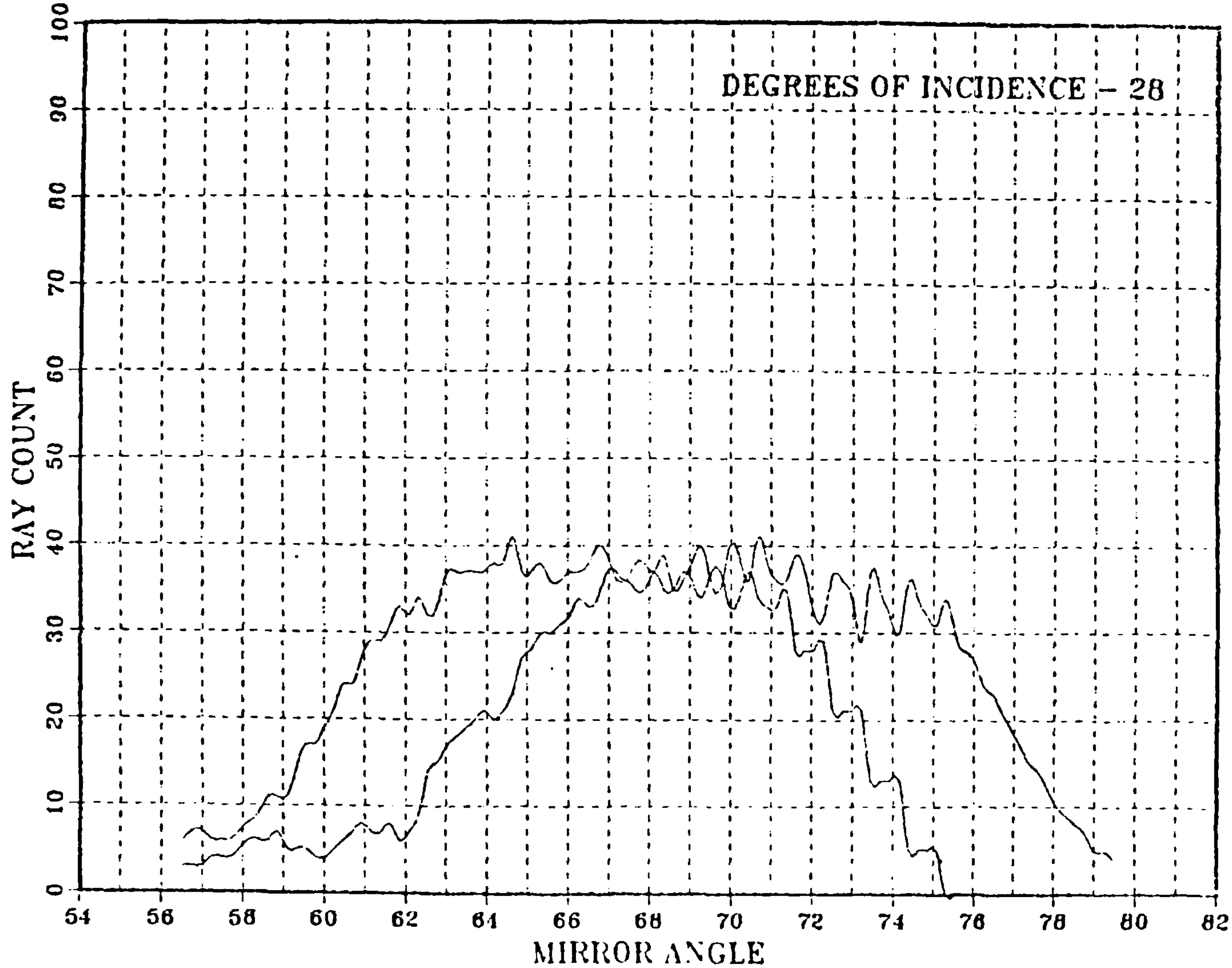

Figure D-7. Detector Signal at 28 Degrees Incidence.

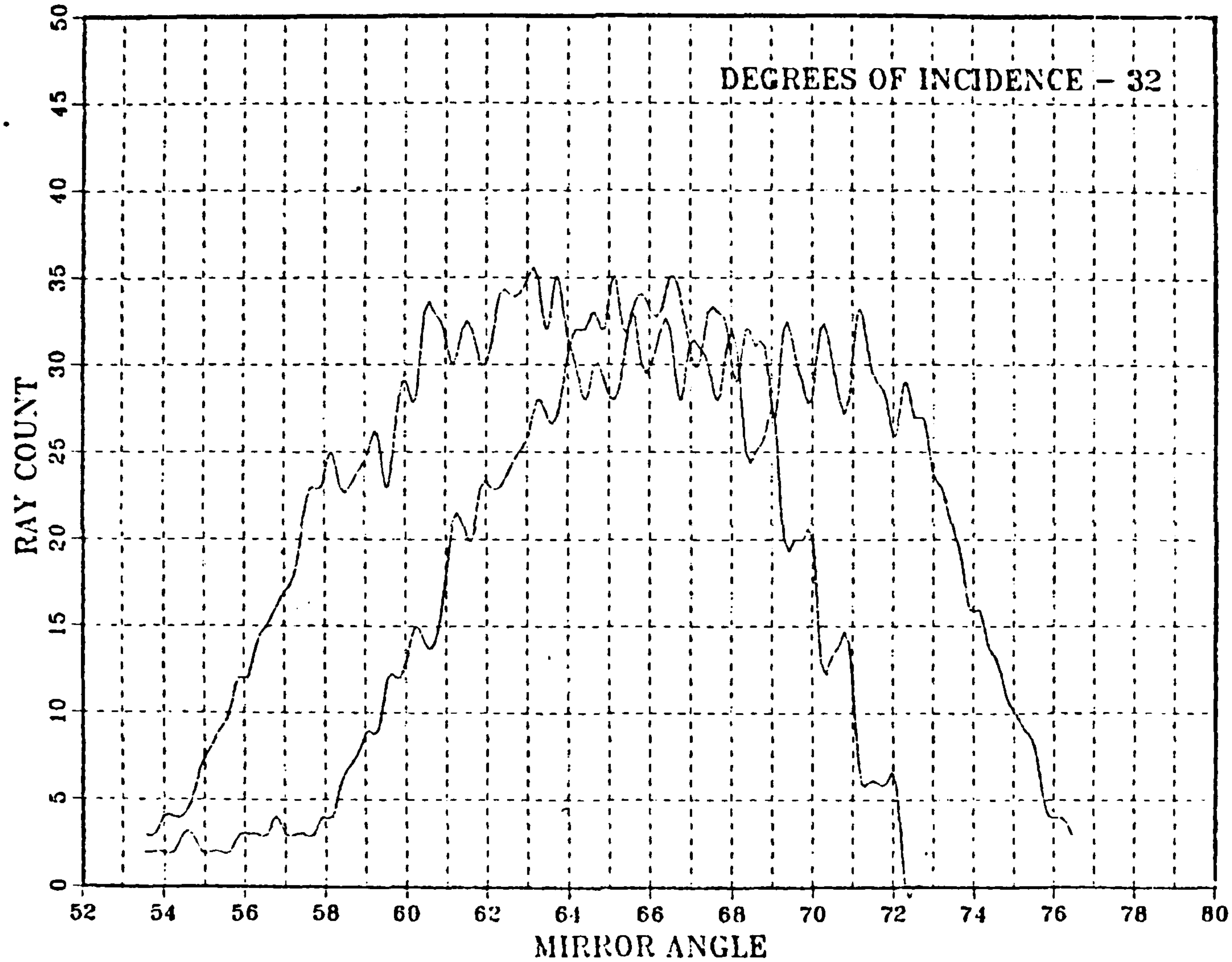

Figure D-8. Detector Signal at 32 Degrees Incidence.

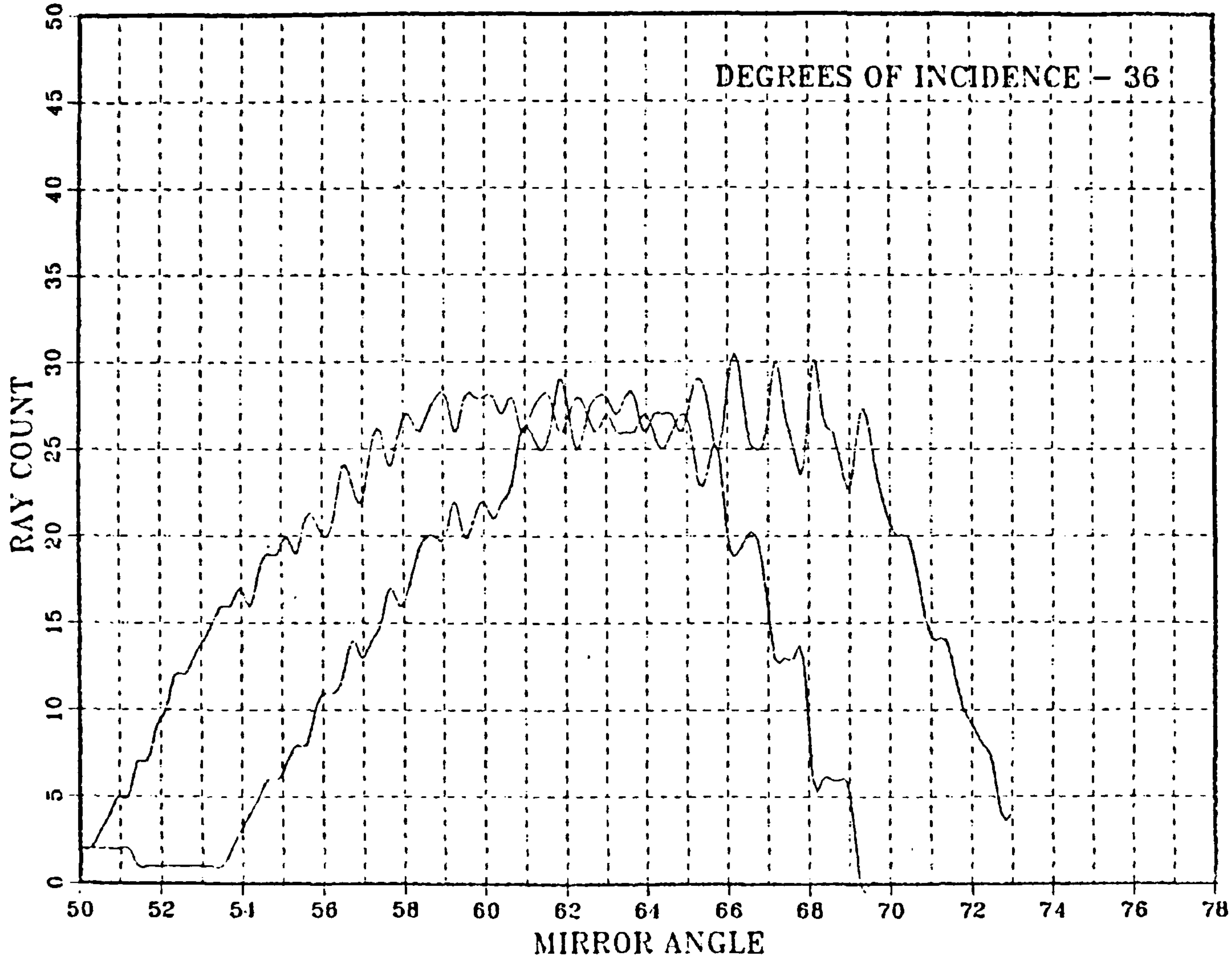

Figure D-9. Detector Signal at 36 Degrees Incidence.

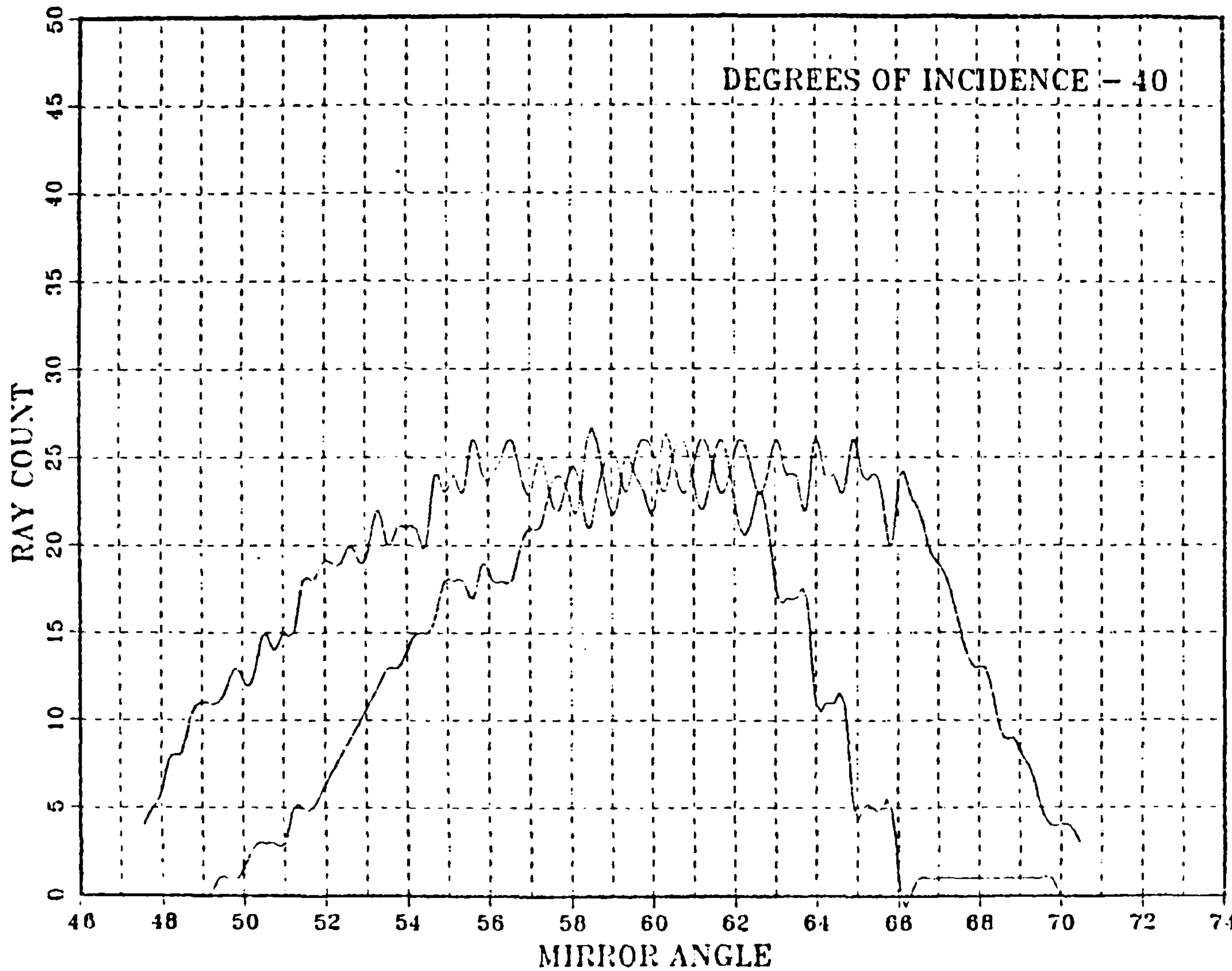

Figure D-10. Detector Signal at 40 Degrees Incidence.

LIST OF REFERENCES

1. Frazier, Robert L., Exterior Ballistics, Guidance Laws and Optics for a Gun-Launched Missile, M. S. Thesis, Naval Postgraduate School, Monterey, California, December 1980.

2. Terrell, James M., Conical Lens for a 5"/54 Gun Launched Missile, M. S. Thesis, Naval Postgraduate School, Monterey, California, June 1981.

3. Amichai, Oded, Sharp Nose Lens Design Using Refractive Index Gradient, Naval Postgraduate School Contractor Report NPS67-82-003CR, Monterey, California, June 1982.

4. Carr, Herbert M., Aerodynamically Efficient Gradient Refractive Index Missile Seeker Lens, M. S. Thesis, Naval Postgraduate School, Monterey, California, October 1982.

5. Marchand, Erich W., Gradient Index Optics, Academic Press, 1978.

6. U. S. Army Missile Command Technical Report RG-CR-82-6, Gradient Index Lens Research-Final Report, by D. T. Moore, p. 36, 19 October 1981.

7. U. S. Army Missile Command Technical Report H8 3-3-BO20-1, Gradient Index Optics Application by R. L. Light, 30 September 1983.

8. D. S. Davidson and A. E. Fuhs, Tracing Nearly Radial Lines in a Spherically Symmetric Gradient Index Lens, Applied Optics, Letter to Editor (in press).

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Directorate of Land Armament and Electronics Engineering and Maintenance Attn: Colonel B. L. Code National Defense Headquarters Ottawa, Ontario, Canada K1A 0K2	2
4.	Distinguished Professor A. E. Fuhs Code 67Fu Department of Aeronautics Naval Postgraduate School Monterey, California 93943	4
5.	Col. R. Larriva, USMC Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209	2
6.	Charles R. Christensen Research Directorate U. S. Army Missile Command (DRSMI-RRO) Redstone Arsenal, Alabama 35898	1
7.	Commander, U. S. Army Missile Command (DRSMI-ROA) Redstone Arsenal, Alabama 35898 Attn: CAPT H. Carr	1
8.	Professor A. E. Milne Code 61Mn Naval Postgraduate School Monterey California 93943	1
9.	Professor D. T. Moore University of Rochester Rochester, New York 14627	1

10. Dr. Lloyd Smith 1
Code 3205
Naval Weapons Center
China Lake, California 93555

11. Directorate of Land Armament and Electronics 2
Engineering and Maintenance
Attn: Capt. D. Davidson
National Defense Headquarters
Ottawa, Ontario, Canada K1A OK2

12. Directorate of Personnel Education 1
and Development (DPED)
National Defense Headquarters
Ottawa, Ontario, Canada K1A OK2